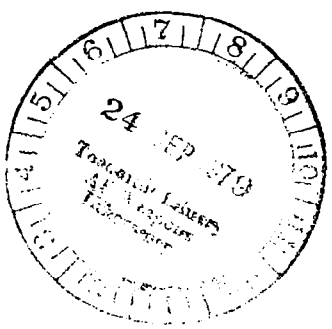
NASA Contractor Report 3167

A Study of the Application of Singular Perturbation Theory

Raman K. Mehra, Robert B. Washburn,
Salim Sajan, and James V. Carroll

CONTRACT NAS1-15113
AUGUST 1979



NASA

NASA Contractor Report 3167

A Study of the Application of Singular Perturbation Theory

Raman K. Mehra, Robert B. Washburn,
Salim Sajan, and James V. Carroll
Scientific Systems, Inc.
Cambridge, Massachusetts

Prepared for
Langley Research Center
under Contract NAS1-15113

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1979

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

This report describes work performed on the development of a hierarchical real-time algorithm for optimal three-dimensional aircraft maneuvers using Singular Perturbation Theory (SPT). New theoretical results justify and develop systematic methods for real-time computation of nonlinear feedback controls by means of SPT and provide an assessment of the accuracy of the resulting SPT control. Practical results apply SPT to obtain a real-time feedback law for the three-dimensional minimum time long range intercept problem for an F-4 aircraft model (six state, three control variable, point mass model). Nonlinear feedback laws are presented for computing the optimal control variables u (throttle), σ (bank angle) and α (angle-of-attack) as a function of target and pursuer aircraft states and desired terminal conditions. A real-time capability assessment of the SPT algorithm on a TI9900 microcomputer has been performed and the control update rates have been determined. The storage and computational requirements of the algorithm are found to be well suited for on-board real-time implementation on a microcomputer.

The accuracy of the SPT solution is analyzed and it is shown how "continuation-type" methods may be used to obtain exact optimal trajectories starting from the SPT solution. The advantage of using predictive terms to supplement the SPT feedback laws is demonstrated for the aircraft trajectory optimization problem. In particular, it is shown that the SPT approximation breaks down near the terminal phase and must be corrected by "continuation" and Generalized Multiple Scale (GMS) methods.

CHAPTER 1
INTRODUCTION

1.1 Summary of Contents

This report describes work performed on the development of a real-time algorithm for optimal three-dimensional aircraft maneuvers using Singular Perturbation Theory (SPT). The optimization problem considered is that of minimum time long range interception. Nonlinear feedback laws are presented for computing the optimal control variables u (throttle), σ (bank angle) and α (angle-of-attack) as a function of target and pursuer aircraft states. A real-time capability assessment of the SPT algorithm on a TI9900 microcomputer has been performed and the control update rates have been determined. The storage and computational requirements of the algorithm are found to be well suited for on-board real-time implementation on a TI9900 microcomputer.

The organization of the report is as follows. Chapter 2 presents the aircraft model (a six-state, point mass approximation) considered in the project. The dynamic equations, constraints and numerical simulation are discussed in detail. Chapter 3 formulates the optimization problem and discusses exact methods of solution--in particular, the continuation method. Chapter 4 presents the general theory of feedback control law computation using SPT. The problems of computational efficiency and accuracy of the SPT method are discussed in detail. Chapter 5 applies these theoretical results to the aircraft optimization problem formulated in Chapter 3. We detail the computational procedures and the necessary approximations made to obtain computationally feasible solutions. Chapter 6 describes the final real-time algorithm and presents numerical examples for various flight trajectories. In addition, we present computation times for the algorithm based on the capabilities of the TI9900 microcomputer. Conclusions are presented in Chapter 7.

1.2 Previous Work on Aircraft Trajectory Optimization

In this section, we trace the historical development of techniques for flight path optimization of high performance aircraft. We will first discuss the minimum-time problem in the vertical plane which has been under consideration

for over 35 years and then discuss the treatment of the horizontal plane problem and the three-dimensional problem which has occurred only recently. Finally, we sketch previous work in aircraft trajectory optimization using singular perturbation theory.

Vertical-plane, minimum-time problem

Before the development of supersonic aircraft, trajectory optimization was performed using the "quasi-steady" approximation in which the accelerations of the aircraft were neglected. Towards the end of World War II with the emergence of higher performance aircraft, this assumption led to results which were less and less accurate. The high acceleration capability of the aircraft particularly along the flight path could no longer be neglected. Kaiser (1944) considered the total energy of the aircraft, expressed as energy height, to obtain a minimum-time climb path which took into account the longitudinal acceleration of the jet interceptor. The solution could be obtained graphically in terms of energy without resorting to the use of the Calculus of Variations. This graphical approach was later used by Lush (1951), Kelley (1952), Fuhrman (1952), Garbell (1953), and Lush (1956) to obtain minimum-time and minimum-fuel paths with free boundary conditions. Rutowski (1954) developed a graphical optimization technique which yielded the theoretical Rutowski energy climb path.

Much of the initial work in obtaining numerical results using the indirect method of the Calculus of Variations was done at RAND around 1949 - 1951. Miele (1950, 1955a, 1955b, 1958, 1962) was active in the field of flight path optimization for over ten years during which he introduced a new method of solving the minimum-time climb problem using Green's theorem. This method is applicable only to a restricted class of problems which can be formulated in a linear form in two transformed variables. Beginning in about 1954, Miele (1959a,b) and Cicala (1955a,b) individually and in collaboration developed the formulation of the fixed end-point, vertical plane problem in terms of the Bolza form of the Calculus of Variations. Here they considered accelerations normal to and along the flight path as well as control inequality constraints and state equality constraints. Numerical results were obtained only for simplified cases. Kelley (1959) studied zoom climbs including consideration of both normal and longitudinal accelerations and discontinuous thrust due to afterburner burnout. Bryson (1966) considered the minimum-time interception of a non-maneuvering

target by utilizing the technique of reducing the state space by the use of dimensionless variables.

In the 1960's much emphasis was placed on the development of gradient algorithms as it was recognized that digital computers were necessary to solve flight path optimization problems. The first successful programs were developed by Bryson (1962) and Kelley (1962). With this technique, an optimum flight path is determined by comparison of an existing trajectory with its predecessors. The method can be made as exact as the model of the aircraft which is being simulated. In general, however, solutions are slow to converge, and will often converge on a local minimum rather than the true minimum. Balakrishnan (1969) proposed a modified gradient approach designed to minimize the large computation times.

Development of numerical techniques for the integration of Euler-Lagrange equations was accomplished by Heerman (1964) and Vincent (1966). The results are generally in the form of a flooded region of trajectories where some refinement is required to arrive at the desired solution. Programs of this type are characterized by instabilities and extreme sensitivity to particular parameters.

There was also a renewal of interest in the energy-state approximation. Boyd and Christie (1965) worked with the concept of energy management and developed operational guidelines without resorting to an indirect method of the Calculus of Variations or gradient solution. Bryson, Desai and Hoffman (1969) presented a fairly complete treatment of the energy-state approximation and applied it to a series of vertical-plane flight path optimization problems to a given range. Other examples of recent applications of the energy-state approximation to the vertical plane problem are by Meier et al. (1970), Schultz and Zagalsky (1972) and Parsons (1972). Parsons considered the minimum-time transit of a supersonic aircraft to a point which is far enough away that there is a central cruise arc at the maximum Mach number of the aircraft.

Sederstrom (1972) presents results of energy management flight tests conducted with an F-8 aircraft. He showed that it was possible to calibrate an individual aircraft and optimize its minimum-time, energy-climb performance on the basis of a relatively simple procedure. Sederstrom et al. (1971) also consider the problem of displays and pilot workload by developing a hybrid simulation of the F-4 aircraft following optimal flight paths obtained using energy

management techniques. Another set of results of flight tests conducted on the F-8D airplane are presented by Capt. Bryan and Allison (1972). Here three flight path schedules were compared; an optimum energy flight path, a flight manual flight path and an optimum energy flight path schedule based on nominal aerodynamic and performance data for the F-8D airplane. More recently, Barman and Erzberger (1976) considered the problem of determining optimum trajectories with a range constraint using the energy-state method for short-haul subsonic aircraft. Uehara, Stewart and Wood (1978) investigate minimum-time loop maneuvers, i.e., maneuvers in the vertical plane in which the flight path angle increases monotonically from 0 to 360 degrees.

Horizontal-plane, minimum-time problem

The minimum-fuel problem was considered in the initial work in determining optimum flight paths in the horizontal plane. Connor (1967) studied the singular arc portion of the minimum-fuel path at constant altitude and Bryson and Lele (1969) presented the full solution of this problem. Final position constraints were not included in this work. Erzberger and Lee (1971) considered the constant-altitude, constant-velocity minimum-time solution to a point and to a line. Hedrick and Bryson (1971a,b) investigated constant-altitude, variable-velocity, minimum-time paths to a final velocity and heading without final horizontal position constraints. Parsons (1972) considers constant-altitude, variable-velocity, minimum-time flight paths to a final point or onto a final line when the flight path is long enough that a cruise period at maximum velocity or a straight bank angle chatter arc is present in the flight. Hoffman and Bryson (1971) considered the case when the cruise period does not exist.

Three-dimensional, minimum-time problem

In 1970, Kelley and Edelbaum (1970) considered three-dimensional, minimum-time flight paths using the energy-state approximation and suggested an asymptotic expansion procedure based on singular perturbation theory to correct the solution near altitude transitions. Horizontal plane final position constraints were not considered and numerical results were not presented. Kelley (1971a) further developed the asymptotic expansion approach suggested above. Numerical results for minimum-time paths without final position constraints were presented by Kelley and Lefton (1972a) and Kelley (1973c).

Hedrick and Bryson (1971, 1972) also treated the three-dimensional,

minimum-time problem without horizontal-plane final position constraints during this same period. Hedrick obtained more complete numerical results than Kelley and associates but considered less realistic flight envelope constraints. Others have considered specific three-dimensional turns with final position constraints without attempting to provide a general characterization of these maneuvers, e.g., Stein et al. (1967), Cambell and Hartsook (1972). Parsons and Bryson (1972) use the energy-state approximation to consider three-dimensional, minimum-time flight paths to a final point or onto a final line when the flight path is long enough that a central cruise period at maximum Mach number is present in the flight. Hoffman and Bryson (1973) extend Parsons' work to consider techniques for real-time on-line optimum flight path control using the reduced-order model obtained from the energy-state approximation. They also studied short-range maneuvers where the cruise period is absent.

Singular Perturbation Theory (SPT) approximation of optimal aircraft trajectories

In a series of papers appearing in the early seventies, Kelley applied the asymptotic expansion methods of singular perturbation theory to aircraft optimization problem. By considering boundary layer correction terms he was able to improve the usual reduced order energy approximation in the regions where instantaneous altitude transitions occur. In the first paper of the series, Kelley and Edelbaum (1970) considered three-dimensional maneuvers, both energy climbs and energy turns. In subsequent papers, Kelley (1970a) considered the general theoretical problem for a two-state system, and Kelley (1970b) applied the method to horizontal plane control of a rocket in a vacuum. The papers, Kelley (1971a) and Kelley and Lefton (1972a), consider energy state models with turn. More generally, Kelley (1971b, 1973c) considers three-dimensional maneuvers with variable mass. Note that Kelley (1973c) gives a detailed account of the singular perturbation approach to aircraft problems and includes most of the earlier work in this paper.

More recently, other investigators have applied asymptotic techniques to aircraft trajectory optimization. Ardema (1976) applied the method of matched asymptotic expansion, one of many singular perturbation methods, to the vertical plane minimum time-to-climb problem. He calculated the zero and first order SPT approximations and compared them to the energy state approximation and the solution obtained by the method of steepest descent (which one could assume to

be optimal). He found that the first-order SPT approximation was close to the steepest descent solution and SPT required much less computation than the latter. In a later paper Ardema (1978) considered a general third order nonlinear SPT problem and studied the occurrence of singular arcs in the solution. Breakwell (1977, 1978) considered the vertical plane, minimum-time problem where drag D is much less than lift L and defined a natural perturbation parameter $\varepsilon = D/L$. He also considered the occurrence of singular arcs in the solution.

Note that the work mentioned so far only applies SPT to off-line aircraft trajectory optimization. In a recent series of papers Calise has applied complete time scale separation to obtain feedback controls by means of SPT. Aggarwal, Calise and Goldstein (1977) consider the vertical plane, minimum-fuel problem for a transport aircraft. Calise (1977a, 1978b) considered the vertical plane minimum-time problem. Calise (1977b) considered feedback control of a missile in the horizontal plane. Calise (1978a) considered the vertical plane problem to minimize a weighted combination of fuel and time for both transport and missile.

This project has emphasized on-line trajectory optimization for aircraft control. The theoretical results (Chapter 4) address the problem of applying SPT to obtain feedback controls which can be computed on-line and stored on-board. We justify and extend the method of complete time scale separation of Calise (1978b) and indicate when the algorithm yields a well-defined control law. (See Subsection 4.3.2 and Appendix 4.2.) We also indicate methods for applying SPT approximations when there is no complete time scale seapration (see Subsections 4.3.3 and 4.3.4), i.e., the use of suboptimal solutions of the slow reduced order and linearization of the fast subproblem around the reduced order solution. Note that when the linearized fast subproblem itself exhibits time scale separation, one can apply the Generalized Multiple Time Scales (GMS) methods of Ramnath and Sandri (1969) to obtain further computational efficiency. Ramnath and Sinhu (1975) applied this method to determine space shuttle re-entry paths requiring minimum mass of heat shielding. In addition, we consider the state space dependence of the accuracy of the SPT approximation and the breakdown of SPT near the terminal target (see Section 4.4).

The practical results (Chapters 5 and 6) apply SPT to obtain a real-time feedback law for the three-dimensional minimum time-to-interception problem for a realistic aircraft model. The six-state point mass model uses real data for

the aerodynamic coefficients and realistic controls and constraints (see Chapter 2). In addition, we obtain the SPT algorithm for this model and assess its real-time capability on a TI9900 microprocessor (see Chapter 6).

CHAPTER 2
AIRCRAFT MODEL

2.1 Introduction

In this chapter we describe the aircraft model used in this project. In summary, this model assumes a point mass approximation--that is, state variables describing the vehicle attitude are either omitted or used as control variables. This assumption implies, of course, that the characteristic time constants of the control system are significantly less than the time constants of the motion; that is, the attitude necessary for generating a certain command force or moment is effectively achieved instantaneously. Another assumption is that the earth's surface is flat, and provides the initial reference system. Next, it is assumed that the vehicle thrust vector is always parallel to the zero lift direction. According to Parsons (1972) this assumption is not a serious restriction. Finally, vehicle mass is assumed constant. The resulting six-state equations of motion are presented in Section 2.2. This section also presents the state and control constraints for the model.

The aircraft whose characteristics were used in this project was an early version of the F-4 used by Bryson, Desai and Hoffman (1969) and as Airplane 1 by Bryson and Parsons (1971). Section 2.3 describes the numerical treatment of the aerodynamic coefficients characterizing this plane. In addition, this section describes the atmospheric model used and the computer simulation of the aircraft dynamics.

2.2 Aircraft Equations of Motion and Constraints

The system of equations for the aircraft presented below is typical of the point mass approximation models encountered throughout the literature. Therefore, the derivation shall not be repeated here. The interested reader may refer to Parsons (1972) from whose work we have chosen our aircraft model.

The point mass model together with our other assumptions mentioned in the introductory Section 2.1 result in a six-dimensional system of first order nonlinear differential equations:

$$\dot{x} = V \cos\beta\cos\gamma \tag{2.1}$$

$$\dot{y} = V \sin\beta\cos\gamma \tag{2.2}$$

$$\dot{h} = V \sin\gamma \tag{2.3}$$

$$\dot{V} = \frac{T\cos\alpha - D - mg\sin\gamma}{m} \tag{2.4}$$

$$\dot{\beta} = \frac{(L + T\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{2.5}$$

$$\dot{\gamma} = \frac{(L + T\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{2.6}$$

In equations (2.1) - (2.6), the state variables are x, y, h, V, β, γ which represent respectively the horizontal position (x,y) in the flat earth's plane, the height h above a fixed ground height, the magnitude V of the velocity, the heading angle β in the horizontal plane of the earth and the flight path angle γ. The reader should refer to Figure 2.1 to see the geometric relation between the state variables. The parameters T, D and L are respectively the thrust, drag and lift forces on the plane. In terms of aerodynamic coefficients these are given as follows:

$$T = u\, T_{max}(M,h) \tag{2.7}$$

where T_{max} is the maximum thrust for height h and Mach number M. Mach number is related to the velocity V by the equation

$$V = c(h)M \tag{2.8}$$

where c(h) is the speed of sound at altitude h.

$$D = D_0 + \eta L_\alpha \alpha^2 \tag{2.9}$$

$$L = L_\alpha \alpha \tag{2.10}$$

$$L_\alpha = C_{L_\alpha} qS \tag{2.11}$$

$$D_0 = C_{D_0} qS \tag{2.12}$$

$$q = \tfrac{1}{2}\rho V^2 \tag{2.13}$$

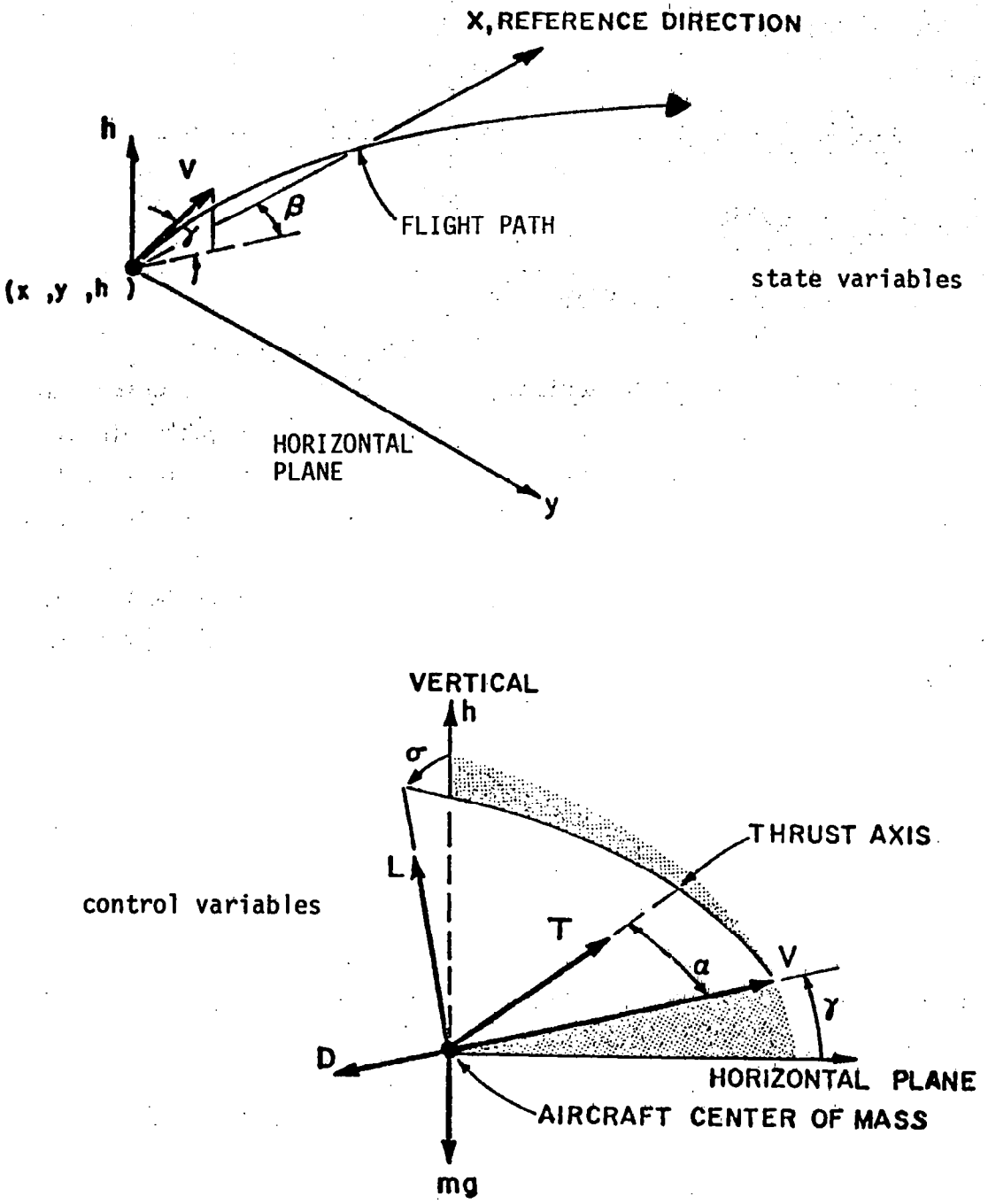


Figure 2.1

Geometric Relationship of State and Control Variables

In the above equations (2.9) - (2.13), C_{L_α} is the lift coefficient, C_{D_0} is the zero lift drag coefficient and η is the aerodynamic efficiency factor. The parameters C_{L_α}, C_{D_0} and η each depend only on Mach number M. The parameter S denotes a characteristic surface area (49.239 m^2 for this aircraft) and $\rho(h)$ is the density at altitude h.

The parameters u, α and σ are considered control variables. The control u represents throttle value and varies between 0 and 1, α denotes angle of attack and σ denotes the bank angle of the plane. See Figure 2.1 for an illustration of the geometric relationship of α and σ to the state variables.

In addition to the dynamic equations (2.1) - (2.6), we have the state and control constraints

$$0 \leq \alpha \leq \alpha_s \tag{2.14}$$

$$0 \leq u \leq 1 \tag{2.15}$$

$$0 \leq V \leq V_{max}(h) \tag{2.16}$$

$$-\sigma_{max} \leq \sigma \leq \sigma_{max} \tag{2.17}$$

$$h_{min} \leq h \tag{2.18}$$

where α_s is the stall value for the angle of attack (assumed 12° in our problem) and σ_{max} is the maximum bank angle (assumed about 76° to correspond to a maximum normal load of 4g during horizontal turns). These constraints represent structural and controllability limitations on the aircraft.

2.3 Numerical Simulation of Aircraft

As we mentioned before in Section 2.1, the aircraft whose characteristics were used in this project is an early version of the F-4 as used by Bryson, Desai and Hoffman (1969) and as Airplane 1 by Bryson and Parsons (1971). The aerodynamic coefficients for this aircraft are tabulated as a function of Mach number and the maximum thrust as a function of Mach number and altitude in Appendix 2.1. The weight of the aircraft was taken as constant at 1.5569×10^5 N and the aerodynamic reference area S is 49.239 m^2. The two atmospheric variables

required in the simulation are sonic speed and air density. These are indicated in Appendix 2.1 as functions of altitude.

The input data describing the atmosphere and the example aircraft as presented above is in tabulated form. A continuous numerical representation of this data is essential for the simulation and the associated partial derivatives are required for the solution of the optimization problem. Therefore, the data was modeled using cubic spline fits which provided continuous values of the data with continuous first and second derivatives. These spline fits are described in more detail as follows:

(a) Sonic Speed c. The sonic speed as presented in Table 3 in Appendix 2.1 is constant above an altitude of 1.2192×10^4 m. Hence a cubic spline fit was constructed for values of height less than the above with an end-condition of zero first derivative at this height. For a value of the first derivative at 0 m, a natural* spline fit was constructed and the value of this derivative obtained was used as the end-condition at 0 m to construct the chosen cubic spline fit.

(b) Atmospheric Density ρ. Since no special end-conditions were required to model atmospheric density, a natural spline fit was constructed.

(c) Aerodynamic Coefficients C_{D_0}, C_{L_α}, η. As in Table 1 of Appendix 2.1, the aerodynamic coefficients are constant for $M \leq 0.8$ which gives end-conditions at $M = 0.8$ of zero first derivatives. The problem of end-conditions at $M = 2.0$ was solved by constructing natural spline fits which provided values for first derivatives at $M = 2.0$ that were used to obtain the required spline fits.

(d) Maximum Thrust T_{max}. Table 2 of Appendix 2.1 shows maximum thrust as a function of two variables: altitude and Mach number. To numerically model this data so as to have available continuous values with partial derivatives, a natural bicubic spline fit was constructed. The natural fit provides zero second partial derivative with respect to Mach number along the altitude boundary, and at the corners zero mixed second partial derivatives. Some interpolated values are presented in Table 4 of Appendix 2.1. Note that the maximum velocity constraint does not allow maximum thrust values at low altitude and high Mach number.

The dynamic simulation of the aircraft was carried out by numerically

*A natural spline fit provides continuous first derivatives and zero second derivatives at the end-points.

integrating the differential equations (2.1) - (2.6) using a second-order Adams-Bashforth integration routine.

APPENDIX 2.1

CHARACTERISTICS OF EXAMPLE AIRCRAFT AND ATMOSPHERE MODEL

Table 1

Aerodynamic Coefficients for F-4 Model as Functions of Mach Number

M	C_{D_0}	C_{L_α}	η
$\leq$ 0.8	0.0130	3.44	0.540
0.9	0.0140	3.58	0.750
1.0	0.0310	4.44	0.800
1.1	0.0388	3.88	0.830
1.2	0.0410	3.44	0.850
1.3	0.0408	3.20	0.875
1.4	0.0390	3.01	0.890
1.5	0.0372	2.84	0.910
1.6	0.0360	2.68	0.920
1.7	0.0354	2.55	0.930
1.8	0.0350	2.44	0.940
1.9	0.0348	2.34	0.945
2.0	0.0346	2.25	0.950

Table 2

Maximum Thrust for F-4 Model as Function of Mach Number and Altitude

h × 10^3 m \ M	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
0	12.60*	13.70	15.35	16.86	16.06	-	-	-	-
1.524	11.21	12.10	13.48	15.26	16.90	16.28	-	-	-
3.048	9.74	10.59	11.83	13.52	15.52	17.13	-	-	-
4.572	8.32	9.12	10.36	11.92	13.92	16.06	17.21	-	-
6.096	7.07	7.70	8.81	10.36	12.14	14.06	15.88	-	-
7.620	5.96	6.54	7.47	8.81	10.50	12.50	14.23	15.39	-
9.144	4.98	5.47	6.27	7.47	8.94	10.77	12.50	13.83	-
10.668	4.05	4.54	5.20	6.18	7.34	8.81	10.40	11.70	12.63
12.192	3.25	3.60	4.18	4.98	5.96	7.21	8.59	9.65	10.45
13.716	2.54	2.89	3.38	3.96	4.76	5.74	6.81	7.70	8.36
15.240	1.96	2.18	2.49	3.02	3.69	4.45	5.29	5.92	6.41
16.764	1.42	1.65	1.87	2.27	2.85	3.43	4.09	4.54	4.85
18.288	0.98	1.11	1.33	1.65	2.05	2.54	3.02	3.38	3.60
19.812	0.58	0.71	0.85	1.07	1.33	1.69	2.05	2.31	2.40
21.336	0.31	0.40	0.49	0.62	0.76	0.98	1.29	1.38	1.42

*all values times 10^4 N

Table 3

Sonic Speed and Density as Functions of Altitude

Altitude	Sonic Speed	Air Density
10^3 m	m/sec	Kgm/m^3
0	340.2	1.225
1.524	334.4	1.055
3.048	328.3	0.9045
4.572	322.2	0.7710
6.096	316.1	0.6525
7.620	309.7	0.5489
9.144	303.2	0.4583
10.668	296.6	0.3968

Altitude	Sonic Speed	Air Density
10^3 m	m/sec	Kgm/m^3
12.192	295.1	0.3015
13.716	295.1	0.2371
15.240	295.1	0.1865
16.764	295.1	0.1466
18.288	295.1	0.1153
21.336	295.1	0.07133
24.384	295.1	0.04410

Table 4

Maximum Thrust Interpolated Using Bicubic Splines

h \ M ×10³m	.4	.5	.6	.7	.8	.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0	12.60*	13.12	13.70	14.46	15.35	16.32	16.86	16.64	16.06	-	-	-	-	-	-	-	-
0.762	11.92	12.37	12.90	13.57	14.41	15.26	16.06	16.64	16.68	-	-	-	-	-	-	-	-
1.524	11.21	11.61	12.10	12.72	13.48	14.28	15.26	16.32	16.90	16.64	16.28	-	-	-	-	-	-
2.286	10.50	10.85	11.34	11.92	12.63	13.43	14.41	15.52	16.37	16.64	16.86	-	-	-	-	-	-
3.048	9.74	10.14	10.59	11.17	11.83	12.63	13.52	14.55	15.52	16.37	17.13	-	-	-	-	-	-
3.810	9.03	9.39	9.86	10.41	11.12	11.88	12.72	13.66	14.72	15.84	16.81	-	-	-	-	-	-
4.572	8.32	8.67	9.12	9.70	10.36	11.12	11.92	12.86	13.92	15.08	16.06	16.68	17.21	-	-	-	-
5.334	7.70	8.01	8.36	8.94	9.56	10.32	11.16	12.05	13.03	14.10	15.03	15.84	16.55	-	-	-	-
6.096	7.07	7.34	7.70	8.18	8.81	9.56	10.36	11.25	12.14	13.08	14.06	14.99	15.88	-	-	-	-
6.858	6.49	6.76	7.07	7.56	8.10	8.81	9.56	10.41	11.30	12.23	13.26	14.23	15.08	-	-	-	-
7.620	5.96	6.23	6.54	6.94	7.47	8.10	8.81	9.61	10.50	11.48	12.50	13.43	14.23	14.81	15.39	-	-
8.382	5.47	5.69	6.01	6.36	6.85	7.43	8.14	8.85	9.74	10.68	11.70	12.59	13.39	14.01	14.63	-	-
9.144	4.98	5.20	5.47	5.83	6.27	6.85	7.47	8.18	8.94	9.83	10.77	11.65	12.50	13.21	13.83	-	-
9.906	4.49	4.72	4.98	5.34	5.74	6.23	6.81	7.43	8.14	8.90	9.79	10.63	11.48	12.19	12.81	-	-
10.668	4.05	4.27	4.54	4.85	5.20	5.65	6.18	6.72	7.34	8.05	8.81	9.61	10.40	11.12	11.70	12.19	12.63
11.430	3.65	3.83	4.05	4.31	4.67	5.07	5.56	6.05	6.63	7.25	7.96	8.72	9.47	10.10	10.63	11.12	11.52
12.192	3.25	3.43	3.60	3.87	4.18	4.54	4.98	5.43	5.96	6.54	7.21	7.92	8.59	9.16	9.65	10.05	10.45
12.954	2.89	3.03	3.25	3.47	3.78	4.09	4.45	4.85	5.34	5.87	6.45	7.07	7.70	8.23	8.67	9.07	9.39
13.716	2.54	2.71	2.89	3.11	3.38	3.65	3.96	4.31	4.76	5.25	5.74	6.27	6.81	7.30	7.70	8.05	8.36
14.478	2.22	2.36	2.54	2.71	2.94	3.20	3.47	3.83	4.18	4.63	5.07	5.56	6.01	6.41	6.76	7.07	7.34
15.240	1.96	2.05	2.18	2.31	2.49	2.71	3.02	3.34	3.69	4.05	4.45	4.89	5.29	5.65	5.92	6.18	6.41
16.002	1.69	1.78	1.91	2.00	2.14	2.38	2.62	2.94	3.25	3.56	3.91	4.31	4.67	4.94	5.20	5.38	5.56
16.764	1.42	1.56	1.65	1.73	1.87	2.05	2.27	2.54	2.85	3.11	3.43	3.78	4.09	4.36	4.54	4.72	4.85
17.526	1.20	1.29	1.38	1.47	1.60	1.73	1.96	2.18	2.45	2.71	2.98	3.29	3.56	3.78	3.96	4.09	4.23
18.288	0.98	1.02	1.11	1.20	1.33	1.47	1.65	1.82	2.05	2.27	2.54	2.80	3.02	3.20	3.38	3.51	3.60
19.050	0.76	0.85	0.89	0.98	1.07	1.20	1.33	1.51	1.69	1.87	2.09	2.31	2.54	2.71	2.85	2.94	2.98
19.812	0.58	0.67	0.71	0.76	0.85	0.93	1.07	1.20	1.33	1.51	1.69	1.87	2.05	2.18	2.31	2.36	2.40
20.574	0.45	0.49	0.53	0.58	0.67	0.76	0.85	0.93	1.02	1.16	1.33	1.51	1.65	1.78	1.82	1.87	1.87
21.336	0.31	0.36	0.40	0.44	0.49	0.53	0.62	0.67	0.76	0.85	0.98	1.16	1.29	1.38	1.38	1.38	1.42

*all values times 10^4 N

CHAPTER 3

THE OPTIMAL CONTROL PROBLEM AND EXACT, ITERATIVE METHODS OF SOLUTION

3.1 Introduction

In this chapter we describe the trajectory optimization problem considered in this project and some exact, iterative methods for its solution. We wish to draw particular attention to the continuation methods described in Subsection 3.3.3. These methods seem to offer very efficient trajectory optimization algorithms, although they are unfortunately not yet fast enough for on-line optimal control of aircraft.

The optimization problem chosen for this problem was the minimum time interception problem. That is, the problem was to minimize the total trajectory time from a given initial point in the six-dimensional state space to a given final point. In Section 3.2 we describe the mathematical formulation of this optimization problem.

3.2 Formulation of the Minimum Time Interception Problem

Here we formulate the minimum time interception problem as in Bryson and Ho (1975) or in Athans and Falb (1966). First, we have a dynamic system

$$\frac{d\underline{x}}{dt} = f(\underline{x},\underline{u}) \tag{2.1}$$

where $\underline{x}$ denotes the state vector and $\underline{u}$ denotes the control variable. We have underlined the state $\underline{x}$ and control $\underline{u}$ to distinguish them from the horizontal position x and the throttle control u. In later sections we will omit the underlines when the context makes clear which case is meant. In our case $\underline{x} = (x,y,h,V,\beta,\gamma)$ and $\underline{u} = (u,\alpha,\sigma)$. Equation (2.1) is the vector representation of the system (2.2.1) - (2.2.6).

In addition to the dynamic equations (2.1) there are inequality constraints

$$C(\underline{x},\underline{u}) \leq \underline{0} \tag{2.2}$$

which must be satisfied at each time of the trajectory. Equation (2.2) is the

vector representation (hence $\underline{0}$ rather than 0) of the state and control constraints given in (2.2.14) - (2.2.18).

At the initial time t_0 an initial state $\underline{x}_0$ is specified,

$$\underline{x}(t_0) = \underline{x}_0 \tag{2.3}$$

where $\underline{x}_0 = (x_0, y_0, h_0, V_0, \beta_0, \gamma_0)$. Similarly, at the final time t_f a final state $\underline{x}_f$ is specified,

$$\underline{x}(t_f) = \underline{x}_f \tag{2.4}$$

where $\underline{x}_f = (x_f, y_f, h_f, V_f, \beta_f, \gamma_f)$. Note that the initial time is fixed but the final time need not be.

Equations (2.1) - (2.4) denote constraints on the possible trajectories of $\underline{x}$, $\underline{u}$. The optimization problem is to minimize the cost of the trajectory where the cost has the form

$$J(\underline{u}) = \int_{t_0}^{t_f} L(\underline{x},\underline{u})\, dt \tag{2.5}$$

For the minimum time problem L is simply the constant,

$$L(\underline{x},\underline{u}) = 1 \tag{2.6}$$

Let us say that a control trajectory $\underline{u}$ is feasible if $\underline{u}$ together with its corresponding state trajectory $\underline{x}$, found from integrating (2.1) with initial conditions (2.3), satisfies the constraints (2.2) and the terminal condition (2.4). The optimal control problem is to find a feasible control trajectory $\underline{u}^*$ such that

$$J(\underline{u}^*) \leq J(\underline{u}) \tag{2.7}$$

for all other feasible controls $\underline{u}$. Solutions of the optimization problem are usually found by solving the Euler-Lagrange first order necessary conditions. These conditions are expressed in terms of a Hamiltonian function defined

$$H(\underline{x},\underline{\lambda},\underline{u}) = L(\underline{x},\underline{u}) + \underline{\lambda}^T f(\underline{x},\underline{u}) \tag{2.8}$$

where $\underline{\lambda}$ denotes a vector of the same dimension as $\underline{x}$ and $\underline{\lambda}^T$ denotes its transpose. In our case, H is given by

$$(2.9) \quad H = 1 + \lambda_x V \cos\beta\cos\gamma + \lambda_y V \sin\beta\cos\gamma + \lambda_h V \sin\gamma + \lambda_V \left[\frac{T\cos\alpha - D - mg\sin\gamma}{m}\right] + \lambda_\beta \left[\frac{(L + T\sin\alpha)\sin\sigma}{mV\cos\gamma}\right] + \lambda_\gamma \left[\frac{(L + T\sin\alpha)\cos\sigma - mg\cos\gamma}{mV}\right]$$

The Euler-Lagrange equations are a system of differential equations in $\underline{x}$ and $\underline{\lambda}$, namely

$$(2.10) \quad \frac{d\underline{x}}{dt} = f(\underline{x},\underline{u})$$

$$(2.11) \quad \frac{d\underline{\lambda}}{dt} = G(\underline{x},\underline{\lambda})$$

For optimal $\underline{u}^*$, $\underline{x}^*$ there is an optimal $\underline{\lambda}^*$ such that (2.10), (2.11) are satisfied and such that

$$(2.12) \quad H(\underline{x}^*,\underline{u}^*,\underline{\lambda}^*) = \min_{\underline{u}} H(\underline{x}^*,\underline{u},\underline{\lambda}^*)$$

at all times. This relationship is known as the minimum principle (see Athans and Falb (1966)).

In principle we can solve (2.12) for $\underline{u}$ in terms of $\underline{x}$, $\underline{\lambda}$ and eliminate $\underline{u}$ from (2.10), (2.11). The result is a system of differential equations in $\underline{x}$ and $\underline{\lambda}$ only,

$$(2.13) \quad \frac{d\underline{x}}{dt} = F(\underline{x},\underline{\lambda})$$

$$(2.14) \quad \frac{d\underline{\lambda}}{dt} = G(\underline{x},\underline{\lambda})$$

with mixed initial and boundary conditions on $\underline{x}$ from (2.3) and (2.4) rather than initial conditions on both $\underline{x}$ and $\underline{\lambda}$. Thus, (2.13), (2.14) with boundary conditions (2.3), (2.4) is a two-point boundary value problem (TPBVP). Many solution techniques solve the original optimization problem by solving this TPBVP. Note that when t_f is not specified, we obtain the extra condition

(2.15) $H(\underline{x}^*,\underline{\lambda}^*,\underline{u}^*) = 0$

at all times. This extra condition can be used to find t_f.

3.3 Exact, Iterative Solutions of the TPBVP

Whether the full system solution or the solution to one of the reduced-order formulations of the problem is being sought, a numerical solution to the TPBVP is necessary. Order reduction is usually achieved through the application of Singular Perturbation Theory (SPT). For a system which is considered "stiff"--that is, the time scales of some of its state variables are significantly faster than those of the remainder--the approximation arising from a reduction in order can be made to be a reasonably good one, if done judiciously.

Assuming, then, that the system has been reduced to a TPBVP, the solution procedures considered in this project fall into three general categories:

(i) Steepest Descent (Gradient) Methods

(ii) Quasilinearization Techniques

(iii) Continuation Methods Using a Parameter

These methods will be discussed in varying detail in the following sections.

3.3.1 Steepest Descent Techniques

These are the most widely used methods, and their strengths and weaknesses are well known. Inequality constraints are typically handled by using penalty functions. There are often convergence problems, due to the presence of state variable inequality constraints and singular arcs. This is because of the absence of control variables in the inequality constraints or in the gradient of the cost function. It is expected that some of these numerical problems may be alleviated by using a "generalized gradient method" as described in Mehra and Davis (1972). Briefly, this method uses the constraints to dictate, at each step, which of the entire set of control and state variables are to be selected as control variables for the next step. The gradient of the cost function with respect to the independent variables, called the generalized gradient, is then computed by solving a set of equations similar to the Euler-Lagrange equations. Directions of search are found using gradient projection and the conjugate gradient method. The procedure, then, is based on the idea that there is no

real mathematical distinction in the use of u(t) or elements of u(t) and x(t) as the independent or manipulative variables of the system $\dot{x} = f(x,u)$. Criteria for selecting the independent set include: any variable which lies on the constraint boundary should be included in the set; choosing some of the state variables as the independent variables often improves the rate of convergence; the independent set must be chosen so as to retain recursiveness to avoid inverting large matrices.

The details of this technique are in Mehra and Davis (1972). It is anticipated that algorithms based on steepest descent, even when applied to the reduced TPBVP's, may be too slow for real time solution of any of the basic problems. This method will be useful, however, in developing the full-system solution.

3.3.2 Quasilinearization Methods

Basically, these methods revolve around doing a linearization around a zero-order solution. Such a solution would arise from setting the "approximation parameter," ε, to zero, as is done in SPT; or, as is done in the case of continuation methods (next section), ε is set to some ε_0, possibly zero, for which the solution is readily obtained. The appeal of quasilinearization techniques lies in the fact that they are far less sensitive to changes in the initial conditions than the shooting continuation methods described in the next section. Thus, in advancing the parameter ε from ε_0 or 0 to its "real" value, generally 1, some combination of techniques based on quasilinearization methodology and continuation theory may be developed. This will hopefully allow the exploitation of the advantages of both schemes, utilizing one where the other is weak. One anticipated disadvantage of the quasilinearization-continuation approach is the computational effort required to step the parameter ε along to its final value.

Basic references on the theory of quasilinearization are Bellman and Kalaba (1965), Dyer and McReynolds (1970), Polak (1971), and Keller (1968). As in Subsection 3.3.1, it is anticipated that inequality constraints can be handled by means of penalty or barrier functions. This should be true both for this section and the following one.

Quasilinearization techniques are cast in the form of Newton-Raphson

problems in Polak (1971). Because of the comparison of this form to the continuation methods of the next section, Polak's algorithms will be presented here. The outline in Dyer and McReynolds (1970) is presented more directly in terms of the calculus of variations problem in optimal control.

For quasilinearization, g is augmented by a third vector element. Suppose we have a differential equation

$$\frac{dx}{dt} = f(x,t) \tag{3.1}$$

with initial and final boundary conditions

$$g_0(x(t_0)) = 0 \tag{3.2}$$

$$g_f(x(t_f)) = 0 \tag{3.3}$$

Note that in the control problem, x in (3.1) would actually include both the state $\underline{x}$ and the adjoint $\underline{\lambda}$. That is, $x = (\underline{x},\underline{\lambda})$ and (3.1) is given by (2.13), (2.14). Let $\tilde{g}(x_0,x(\cdot))$ be the function defined by

$$\tilde{g}(x_0,x(\cdot))(t) = x_0 + \int_{t_0}^{t} f(x(s),s)ds - x(t) \tag{3.4}$$

and define g from g_0, g_f and $\tilde{g}$ so that

$$g(x_0,x(\cdot)) = \begin{bmatrix} \tilde{g}(x_0,x(\cdot)) \\ g_0(x(t_0)) \\ g_f(x(t_f)) \end{bmatrix} \tag{3.5}$$

Thus, g maps the pair $(x_0,x(\cdot))$ into the pair $(y_0,y(\cdot))$ where $y_0 = (g_0(x(t_0)), g_f(x(t_f)))$ and $y(t) = \tilde{g}(x_0,x(\cdot))(t)$. In the following, let L denote the linear space of pairs $(x_0,x(\cdot))$ where x_0 is a vector and $x(\cdot)$ is a piecewise continuously differentiable function of t. Then g is a continuously differentiable map of L into itself. We will assume that $[\partial g(z)/\partial z]^{-1}$ exists for all z in a sufficiently large subset of L, where $z = (x_0,x(\cdot))$. With this assumption, the Newton-Raphson method for solving $g(z) = 0$ is defined by

$$(3.6) \qquad \frac{\partial g(\overset{j}{z})}{\partial z}\,(\overset{j+1}{z} - \overset{j}{z}) = -g(\overset{j}{z})$$

where $\overset{j}{z}$ is the $j^{\underline{th}}$ iteration for the solution. Thus, $\overset{j+1}{z}$ is found from $\overset{j}{z}$ by utilizing the above relationship, exactly as in the finite-dimensional case (next section). Due to the presence of $x(t) \in C^1_{2N}[t_0,t_f]$ in z, the Newton-Raphson method above has been formulated in a Banach space L.

Substituting the g into the above equation, rearranging terms, and differentiating the terms associated with $\tilde{g}$ with respect to time, there results

$$(3.7) \qquad \frac{d\overset{j+1}{x}(t)}{dt} = \frac{\partial f(\overset{j}{x}(t),t)}{\partial x}\,[\overset{j+1}{x}(t) - \overset{j}{x}(t)] + f(\overset{j}{x}(t),t), \qquad t \in [t_0,t_f]$$

$$g_0(\overset{j+1}{x}(t_0)) = 0, \quad g_f(\overset{j+1}{x}(t_f)) = 0$$

This differential system is called the quasilinearization version of the Newton-Raphson method. McGill and Kenneth (1963) and Bellman and Kalaba (1965) provide more details. The algorithm proceeds as follows:

1) Select an $\overset{0}{x}(\cdot) \in C^1_{2N}[t_0,t_f]$ such that $g_0 = g_f = 0$; if x_0 and x_f are such that the boundary conditions are met, then an acceptable $\overset{0}{x}(t)$ may be
$$\overset{0}{x}(t) = x_0 + [(t-t_0)/(t_f-t_0)](x_f-x_0), \qquad t \in [t_0,t_f];$$
2) Set $j = 0$;
3) For $t \in [t_0,t_f]$, compute $f(\overset{j}{x}(t),t)$ and $\frac{\partial f}{\partial x}(\overset{j}{x}(t),t)$;
4) Compute $\overset{j+1}{x}(t)$ by solving the Newton-Raphson differential system, integrating in a stable direction (or else, combining the technique of Roberts and Shipman (1967), described in the next section);
5) If $\overset{j+1}{x}(t)$ is "close to" $\overset{j}{x}(t)$ by some standard, stop;
6) Otherwise, set $j = j+1$ and go to step (3).

Quasilinearization techniques such as the one outlined above provide at each step an approximation to a solution of $\dot{x} = f(x(t),t)$ which satisfies the boundary conditions. This is a major difference between quasilinearization techniques and procedures outlined below. The latter, at each step, provide an approximation to the solution and the boundary conditions which satsify the differential equation.

3.3.3 Continuation Methods for the Solution of the TPBVP

In this section we are interested in describing some continuation or imbedding methods for solving the two-point boundary value problem and discussing their advantages over more conventional methods. To this end, we have omitted many mathematical details necessary for a logically rigorous presentation of this material. We hope this omission will make the presentation clearer for the reader unfamiliar with continuation methods. Mathematically rigorous results may be found in Ortega and Rheinboldt (1970).

For definiteness, consider the following two-point boundary value problem (TPBVP) which depends on a scalar parameter:

TPBVP: Find $x(t,\varepsilon)$, $a(\varepsilon)$ and $\tau(\varepsilon)$ for times t and all parameters ε such that $x(t,\varepsilon)$, $a(\varepsilon)$ are vectors in R^n and $\tau(\varepsilon)$ is a scalar time, and such that these quantities satisfy

$$\frac{\partial x}{\partial t}(t,\varepsilon) = f((t,\varepsilon)t,\varepsilon) \tag{3.8}$$

$$x(0,\varepsilon) = a(\varepsilon) \tag{3.9}$$

$$\phi(a(\varepsilon),x(\tau(\varepsilon),\varepsilon),\tau(\varepsilon),\varepsilon) = 0 \tag{3.10}$$

for all ε and all t with $0 \le t \le \tau(\varepsilon)$.

Note that the function ϕ, which maps $R^n \times R^n \times R^1 \times R^1$ into R^{n+1}, is given beforehand, and equation (3.10) summarizes the n+1 initial and final conditions necessary to deduce n initial conditions (the vector $a(\varepsilon)$) and the terminal time $\tau(\varepsilon)$. One can choose ϕ so that (3.8) - (3.10) represent almost any initial or boundary value problem. In particular, (3.8) - (3.10) can model the TPBVP for which the terminal time τ is not given explicitly (this is the situation for minimum time control problems).

To solve the TPBVP we transcribe the equations (3.8) - (3.10) to a system of n+1 nonlinear equations for the initial condition $a(\varepsilon)$ and the final time $\tau(\varepsilon)$. To make this transcription define the function $x(t;a,\varepsilon)$ as the solution of the initial value problem

$$\frac{\partial x}{\partial t}(t;a,\varepsilon) = f(x(t;a,\varepsilon),t,\varepsilon) \tag{3.11}$$

$$x(0;a,\varepsilon) = a \tag{3.12}$$

for all t, a, ε. Then we solve for $a(\varepsilon)$ and $\tau(\varepsilon)$ from the equation

$$\phi(a(\varepsilon),x(\tau(\varepsilon);\ a(\varepsilon),\varepsilon),\tau(\varepsilon),\varepsilon) = 0 \tag{3.13}$$

Note that (3.13) represents n+1 equations which we desire to solve for the initial state $a(\varepsilon)$ (n conditions) and the final time $\tau(\varepsilon)$ (one more condition). The TPBVP of (3.8) - (3.10) has a solution if and only if (3.13) has a solution. Thus, the TPBVP is reduced to solving the nonlinear equation (3.13).

Different techniques for solving the TPBVP derive from techniques for solving the equation (3.13). Thus, let us define the function $G(a,\tau,\varepsilon)$ as

$$G(a,\tau,\varepsilon) = \phi(a,\ x(\tau;a,\varepsilon),\ \tau,\ \varepsilon) \tag{3.14}$$

and let us generically represent (a,ε) by the n+1 vector ν. Then (3.13) takes the more general form

$$G(\nu,\varepsilon) = 0 \tag{3.15}$$

which we solve for ν as a function of ε.

Let us suppose that ε varies between 0 and 1. Often our problem is to solve a difficult problem

$$G(\nu,1) = 0 \tag{3.16}$$

when we know how to solve an easier problem

$$G(\nu,0) = 0 \tag{3.17}$$

Sometimes the parameter ε occurs naturally in the problem (i.e., the viscosity in a hydrodynamic problem), but often we introduce the parameter ε artificially. In either case, the rationale for replacing the single equation (3.16) with a family of equations (3.15) is that we may be able to continue the solution at $\varepsilon = 0$ in (3.17) to the solution at $\varepsilon = 1$ in (3.16) more easily than computing the solution at $\varepsilon = 1$ in (3.16) by itself. For example, a classic technique for continuing the solution $\nu(0)$ of (3.17) to a solution $\nu(\varepsilon)$ of (3.15) for $\varepsilon > 0$ is to expand the vector function $\nu(\varepsilon)$ in a Taylor series in ε. The validity of the expansion requires some regularity of the function G with respect to ε, but in some cases where G does not depend regularly on ε it is often still possible to

use perturbation analysis to find a singular perturbation expansion of $\nu(\varepsilon)$ in terms of ε.

As a simple but concrete example, consider the following two quadratic equations:

$$y + y^2 = \varepsilon \tag{3.18}$$

and

$$x + \varepsilon x^2 = 1 \tag{3.19}$$

The solution at $\varepsilon = 0$ of equation (3.18) gives $y(0) = 0$ or $y(0) = -1$. The solution $y(\varepsilon)$ of (3.18) for $\varepsilon > 0$ depends regularly on ε and has the two possible power series expansions

$$y(\varepsilon) = \varepsilon - \varepsilon^2 + \ldots \tag{3.20}$$

or

$$y(\varepsilon) = -1 - \varepsilon + \varepsilon^2 + \ldots \tag{3.21}$$

This series corresponds to the Taylor expansion in powers of ε of

$$y(\varepsilon) = \frac{-1 + \sqrt{1 + 4\varepsilon}}{2} \tag{3.22}$$

and

$$y(\varepsilon) = \frac{-1 - \sqrt{1 + 4\varepsilon}}{2} \tag{2.23}$$

Equation (3.19) is not regular with respect to ε as one can see by setting $\varepsilon = 0$; there is only one solution $x(0) = 1$, whereas there must be two whenever $\varepsilon \neq 0$. Nevertheless, one can still expand the solution $x(\varepsilon)$ of (3.19) in the singular perturbation series

$$x(\varepsilon) = 1 - \varepsilon + \ldots \tag{3.24}$$

or

$$x(\varepsilon) = -\frac{1}{\varepsilon} - 1 + 2 + \ldots \tag{3.25}$$

which series correspond to the exact solutions

(3.26) $$x(\varepsilon) = \frac{-1 + \sqrt{1 + 4\varepsilon}}{2\varepsilon}$$

and

(3.27) $$x(\varepsilon) = \frac{-1 - \sqrt{1 + 4\varepsilon}}{2\varepsilon}$$

respectively.

The approximations (3.20), (3.21) to (3.18) or (3.24), (3.25) to (3.19) are good when ε is "small," but the trouble is that $\varepsilon = 1$ is not small. In fact, the expansions (3.20), (3.21) and (3.24), (3.25) are valid only when $|\varepsilon| < \frac{1}{4}$. For $|\varepsilon| > \frac{1}{4}$, the series do not converge and one cannot use finitely many terms of the series to approximate the exact answer. For example, if one tries to use the series from (3.20) for $\varepsilon = 1$ to approximate (3.22), one obtains the approximations $y(\varepsilon) \simeq 0,1,-1,2,-5,14,-42,132$ and so on, which become progressively worse as one adds more terms to the series in (3.20). The exact answer is (3.22) with $\varepsilon = 1$, which gives $y(\varepsilon) = .618033989....$

In our examples in (3.18) or (3.19) perturbation analysis cannot continue the $\varepsilon = 0$ solution beyond $\varepsilon = \frac{1}{4}$. Nevertheless, there are continuation methods which can continue the $\varepsilon = 0$ solution all the way to $\varepsilon = 1$. One such method is the method of differentiation with respect to a parameter of Davidenko (1953).

Consider the general equation (3.15), $G(\nu,\varepsilon) = 0$, and suppose that there is a solution $\nu(\varepsilon)$ of (3.15) which is continuously differentiable with respect to ε. Taking the derivative of equation (3.15) with respect to ε gives us the following differential equation for ν:

(3.28) $$G_\nu \frac{d\nu}{d\varepsilon} + G_\varepsilon = 0$$

In (3.28) the expression G_ν is the matrix of first partial derivatives of G with respect to ν, and G_ε is the vector of first partial derivatives with respect to ε. The initial conditions for the differential equation (3.28) is just the $\nu(0)$ given by equation (3.17), namely

(3.29) $$G(\nu(0),0) = 0$$

Assuming that we can solve the linear equation

(3.30) $G_\nu(\nu,\varepsilon)w + G_\varepsilon(\nu,\varepsilon) = 0$

for w given ν and ε, then we can numerically integrate equation (3.28) from the initial condition (3.29). If the matrix $G_\nu(\nu,\varepsilon)$ becomes singular and (3.30) has no solution, then the numerical integration of (3.28) may fail. However, Keller (1977) presents methods for continuing the integration when $G_\nu(\nu,\varepsilon)$ becomes singular. In this case one finds that (3.28) has bifurcating solutions. We discuss such bifurcations further in Appendix 3.1.

As a simple example consider equation (3.28) again. In this case (3.28) becomes

(3.31) $(1 + 2y)\frac{dy}{d\varepsilon} = 1$

with two possible initial conditions, either $y(0) = 0$ or $y(0) = -1$. For all $\varepsilon \geq 0$, we can integrate (3.31) without difficulty and obtain the two solutions

(3.32) $y(\varepsilon) = \frac{-1 + \sqrt{1 + 4\varepsilon}}{2}$

from the initial condition $y(0) = 0$, and

(3.33) $y(\varepsilon) = \frac{-1 - \sqrt{1 + 4\varepsilon}}{2}$

from the initial condition $y(0) = -1$.

Note that Davidenko's method succeeds in finding the solutions at $\varepsilon = 1$ whereas the power series method in (3.20), (3.21) fails. Using an integration step size of $h = .1$ and Euler's method of integrating (3.31), we obtain the approximation .6372 for $y(1)$ corresponding to $y(0) = 0$. Compare this to the exact solution $y(1) = .6180....$

The Newton-Raphson method is an alternative for solving the equation (3.16), $G(\nu,\varepsilon) = 0$, directly for any particular ε, provided one has an initial estimate of $\nu(\varepsilon)$. Recall that the Newton-Raphson method calculates successive approximations $\nu_k(\varepsilon)$ from the recursive formula,

(3.24) $\nu_{k+1}(\varepsilon) = \nu_k(\varepsilon) - G_\nu(\nu_k(\varepsilon),\varepsilon)^{-1}G(\nu_k(\varepsilon),\varepsilon)$

To start the method one requires an initial estimate $\nu_1(\varepsilon)$. In essence, if $\nu_1(\varepsilon)$ is "reasonably close" to $\nu(\varepsilon)$, then the estimates $\nu_k(\varepsilon)$ computed from (3.34) converge very quickly to $\nu(\varepsilon)$ as k tends to infinity. The disadvantage of using Newton-Raphson's method is that we may not have an initial ν_1 which is reasonably close to the actual solution ν. A continuation method such as Davidenko's method can help overcome this difficulty.

As a specific example, consider the trivial equation

$$1 - e^{-(\nu-\varepsilon)} = 0 \tag{3.35}$$

and suppose that we know that $\nu(10) = 10$ but we wish to approximate $\nu(0)$ from this initial guess. The guess $\nu_1(0) = 10$ is disastrous for applying Newton-Raphson's method. The recursion equation (3.34) for $\varepsilon = 0$ becomes

$$\nu_{k+1} = \nu_k + 1 - e^{\nu_k} \tag{3.36}$$

and we find that if $\nu_1 = 10$, then $\nu_2 \simeq -22015$, $\nu_3 \simeq -22014$, $\nu_4 \simeq -22013$ and so on. One requires over 22,000 iterations of (3.36) to approach the true root $\nu = 0$.

Davidenko's method leads to the numerical integration of the differential equation

$$\frac{d\nu}{d\varepsilon} = 1 \tag{3.37}$$

with the initial condition $\nu(10) = 10$. If we use Euler's method of integration with step size h, we will require roughly $10 \times h^{-1}$ integration steps to reach the approximation of $\nu(0)$. Note that the approximation will be accurate to order h. Thus, Davidenko's method leads to the approximate solution of (3.35) at $\varepsilon = 0$ in a reasonably few integration steps if we do not require great accuracy in our approximation.

For future reference we now present another continuation method different from Davidenko's method. Ortega and Rheinboldt (1970) present this method as an example of continuation methods.

This continuation method differs from Davidenko's method in that it does not use a differential equation to calculate $\nu(\varepsilon)$ at successive values of ε. Instead it uses the Newton-Raphson method (3.34) in a recursive fashion as we

now explain.

Let us solve $G(\nu,\varepsilon) = 0$ at $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ where $0 < \varepsilon_1 < \varepsilon_2 < \varepsilon_3 \ldots \varepsilon_n$. Suppose that we have an initial estimate ν^0 for $\nu(0)$ or perhaps we know $\nu^0 = \nu(0)$. Also assume given the sequence $m_1, m_2, \ldots, m_n$ of positive integers. From this initial estimate one defines $\nu_k(\varepsilon_i)$ recursively as follows. For $1 \le k \le m_i - 1$, define $\nu_{k+1}(\varepsilon_i)$ as

$$\nu_{k+1}(\varepsilon_i) = \nu_k(\varepsilon_i) - G_\nu(\nu_k(\varepsilon_i),\varepsilon_i)^{-1} G(\nu_k(\varepsilon_i),\varepsilon_i) \tag{3.38}$$

for $k = 1$ and $i \ge 1$, define $\nu_1(\varepsilon_i)$ as $\nu_{m_{i-1}}(\varepsilon_{i-1})$. This technique operates by recursively calculating Newton-Raphson approximations of the equation $G(\nu,\varepsilon_i) = 0$ by using as an initial estimate for $\nu(\varepsilon_i)$ the approximation of $\nu(\varepsilon_{i-1})$ found from solving $G(\nu,\varepsilon_{i-1}) = 0$. By suitably choosing the ε_i and the m_i, one can continue a solution of $G(\nu,0) = 0$ to a solution of $G(\nu,\varepsilon) = 0$ for relatively large values of ε. Accuracy depends on how close together one takes the ε_i and how large one takes the m_i.

The Newton-Raphson technique is a fast converging approximation method provided that the initial estimate is close to the exact solution. In this case, the error at each step is reduced by squaring. That is, the error δ_{k+1} at the k+1 iteration is approximately δ_k^2, and thus, convergence is very fast. A continuation method such as Davidenko's method is inferior to Newton-Raphson's method from the standpoint of accuracy with respect to computation speed. The Davidenko method makes a final error proportional to the integration step size h in the integration of (3.28) (or proportional to h^α for $\alpha > 1$ if a better integration scheme than Euler's method is employed). If one tries to obtain good accuracy by choosing h small, the integration time may be quite long. For example, numerically integrating (3.37) to obtain the solution $\nu(0)$ of (3.35) to four decimal places accuracy would require about 10^5 integration steps using Euler's method of integration.

However, although the continuation methods are inefficient in computing highly accurate solutions, these methods are much less sensitive than Newton-Raphson to the initial estimate of the solution.

Similarly, continuation methods are not as efficient as perturbation expansions when the perturbation parameter is small. On the other hand, continuation

methods such as Davidenko's method provide solutions even when the perturbation parameter is large.

Ideally, one might use a continuation method together with a method such as Newton-Raphson or perturbation analysis. For example, one might first use Davidenko's method with a modest integration step size to compute an approximation reasonably close to the exact solution. Then, using Newton-Raphson's method and using the Davidenko approximation as an initial estimate, one could obtain a very accurate approximation of the exact solution. Such an algorithm is described in Appendix 3.2.

Having presented these different continuation techniques for computing the solution of $G(\nu,\varepsilon) = 0$, we now show how to apply these techniques to the TPBVP. There are basically two continuation methods for solving the TPBVP which correspond to the two continuation methods for solving the equation $G(\nu,\varepsilon) = 0$. Kubicek and Hlavacek (1973) use Davidenko's method to solve the TPBVP, and Roberts and Shipman (1967) use the continuation version of Newton-Raphson to solve TPBVP.

Kubkcek and Hlavacek Algorithm

They also call this method general parameter mapping or GPM. The function G is given by equation (3.14) and then the TPBVP is solved by solving (3.15). Kubicek and Hlavacek do this by using Davidenko's method--that is, by solving the differential equation (3.28). To use Davidenko's method, we must compute G_ν and G_ε as follows:

$$(3.39) \qquad G_\nu(a,\tau,\varepsilon) = \begin{pmatrix} \phi_a + \phi_x \cdot x_a \\ \phi_x \cdot x_\tau + \phi_\tau \end{pmatrix}$$

and

$$(3.40) \qquad G_\varepsilon(a,\tau,\varepsilon) = \phi_x \cdot x_\varepsilon + \phi_\varepsilon$$

The subscripts in equations (3.39) and (3.40) denote partial derivatives. Thus, $\phi_a = \frac{\partial \phi}{\partial a}$ is the $(n+1) \times n$ matrix of first partial derivatives of $\phi(a,x,\phi,\varepsilon)$ with respect to a (remember that ϕ takes its values in R^{n+1} and a is a vector of R^n). Similarly, ϕ_x is the $(n+1) \times n$ matrix of derivatives of ϕ with respect to x; x_a is the $n \times n$ matrix of derivatives of $x(\tau;a,\varepsilon)$ with respect to a. The expressions x_τ and x_ε denote the n-vectors $\frac{\partial x}{\partial \tau}$ and $\frac{\partial x}{\partial \varepsilon}$. Likewise, ϕ_τ and ϕ_ε denote the $(n+1)$-vectors $\frac{\partial \phi}{\partial \tau}$ and $\frac{\partial \phi}{\partial \varepsilon}$.

From the original formulation of the problem, one can find ϕ, ϕ_x, ϕ_a, ϕ_τ and ϕ_ε. However, to compute G_ν and G_ε, one needs x, x_a, x_τ and x_ε, and one must integrate (3.11), (3.12) directly. That is, one integrates (3.11), (3.12) to $t=\tau$ to obtain $x(\tau;a,\varepsilon)$. Then $x_\tau(\tau;a,\varepsilon)$ is given from (3.11) as

$$x_\tau = f(x(\tau;a,\varepsilon),\tau,\varepsilon) \tag{3.41}$$

To obtain x_a and x_ε we differentiate (3.11), (3.12) with respect to a and ε respectively. Thus, x_a is the solution of the initial value problem

$$\frac{\partial x_a}{\partial t} = f_x(x(t;a,\varepsilon),t,\varepsilon)\cdot x_a \tag{3.42}$$

$$x_a(0;a,\varepsilon) = I \tag{3.43}$$

where I is the $n\times n$ identity matrix. Likewise, x_ε is the solution of the initial value problem

$$\frac{\partial x_\varepsilon}{\partial t} = f_x(x(t;a,\varepsilon),t,\varepsilon)\cdot x_\varepsilon + f_\varepsilon(x(t;a,\varepsilon),t,\varepsilon) \tag{3.44}$$

$$x_\varepsilon(0;a,\varepsilon) = 0 \tag{3.45}$$

Thus, an algorithm for solving the TPBVP (3.8), (3.9), (3.10) for $0\le\varepsilon\le 1$ might be the following:

1. Solve the problem for $\varepsilon = 0$. Set $\varepsilon < 0$ and $a(\varepsilon) = a_0$, $\tau(\varepsilon) = \tau_0$. Choose integration step size h (for integration with respect to ε).
2. For ε, $a(\varepsilon)$, $\tau(\varepsilon)$ given, compute x, x_a, x_τ and x_ε from integrating (3.11), (3.12), (3.41), (3.42) - (3.45).
3. With these values of ε, $a(\varepsilon)$, $\tau(\varepsilon)$ and x, x_a, x_τ, x_ε from #2, use (3.39) and (3.40) to compute G_τ and G_ε.
4. Using some numerical integration scheme on (3.28), compute $\frac{d\nu}{d\varepsilon} = \left(\frac{da}{d\varepsilon}, \frac{d\tau}{d\varepsilon}\right)$ and use this to evaluate $(a(\varepsilon+h), \tau(\varepsilon+h))$.
5. Update ε to $\varepsilon+h$. If $\varepsilon\ge 1$, stop. If $\varepsilon < 1$, return to step 2.

Roberts and Shipman

Roberts and Shipman (1967, 1968) solve a TPBVP

(3.46) $\frac{dx}{dt} = f(x,t), \quad 0 \le t \le t_1$

(3.47) $x(0) = a$

(3.48) $x(a,x(t_1)) = 0$

where t_1 is given by allowing the terminal time to vary from 0 to t_1 and keeping all other conditions the same. In terms of the general formulation (3.8), (3.9), (3.10), Roberts and Shipman solve the problem

(3.49) $\frac{\partial x}{\partial t} = f(x(t;\varepsilon),t)$

(3.50) $x(0,\varepsilon) = a$

(3.51) $x(a,x(\tau(\varepsilon),\varepsilon)) = 0, \quad \tau(\varepsilon) = \varepsilon$

for all ε and all t such that $0 \le t \le \tau(\varepsilon) = \varepsilon$. Thus, Roberts and Shipman allow just the final time τ to be the parameter ε, and they solve the original problem (3.46) - (3.48) by varying ε from 0 to t_1 in (3.49) - (3.51). The continuation method they use is the extension of Newton-Raphson's method we described for solving $G(\nu,\varepsilon) = 0$. Since this continuation method is easily applied to the more general problem (3.8) - (3.10), we do this rather than treat the specific problem Roberts and Shipman (1967, 1968) use in their papers. Nevertheless, for future reference we will refer to this continuation solution of the general TPBVP as the Roberts-Shipman method.

The method consists of applying the equation (3.38) to solve $G(\nu,\varepsilon) = 0$ when G is defined by (3.14). One obtains G_ν from (3.39) just as in using Kubicek and Hlavacek's GPM method. This requires numerical integration of the equations (3.11), (3.12) and (3.41) - (3.43) just as before. One possible algorithm to solve the TPBVP using the Roberts-Shipman method is then the following:

1. Solve the problem for $\varepsilon = 0$ to find a_0, τ_0. Choose $\varepsilon_0 = 0$, $0 < \varepsilon_1 < \varepsilon_2 \ldots \varepsilon_n$, and choose positive integers m_i; $1 \le i \le n$. Set $i = 0$, $k = 0$, $a_k(\varepsilon_i) = a_0$, $\tau_k(\varepsilon_i) = \tau_0$.
2. For ε_i, $a_k(\varepsilon_i)$, $\tau_k(\varepsilon_i)$ given, compute x, x_a, x_τ from (3.11), (3.12), (3.41) - (3.43).
3. With these values of ε_i, $a_k(\varepsilon_i)$, $\tau_k(\varepsilon_i)$ and x, x_a, x_τ from #2, use (3.39) to compute G_ν.

4. Invert G_ν from #3 and evaluate $(a_{k+1}(\varepsilon_i), \tau_{i+1}(\varepsilon_i)) \equiv \nu_{k+1}(\varepsilon_i)$ from equation (3.38) (the Newton-Raphson step).
5. Update k to k+1. If $k = m_i$, go on to step 6; otherwise return to step 2.
6. If $i = n$, stop. Otherwise set $a_0(\varepsilon_{i+1}) = a_k(\varepsilon_i)$ and $\tau_0(\varepsilon_{i+1}) = \tau_k(\varepsilon_i)$ and reset k to 0; set i to i+1 and return to step 2.

Summary of History and Advantages of Continuation Methods

Continuation or imbedding methods have long been used to prove existence theorems for operator equations. Ortega and Rheinboldt (1970) give a nice discussion of the method with many historical notes and references. Ficken (1951) contains references and notes on the literature previous to 1950, including some from the last century.

Lahaye (1934) and later, independently, Davidenko (1953) first applied continuation methods to the numerical solution of nonlinear equations of the form $G(\nu,\varepsilon) = 0$. Davidenko introduced the method of differentiating $G(\nu,\varepsilon)$ with respect to the parameter ε to obtain a differential equation for ν.

More recently, the method of continuation has been used to solve numerically fixed point problems. See Kellogg, Li and Yorke (1976) for example. Rigorous and powerful mathematical treatments of the continuation method rely on topological homotopy theory.

A preliminary study of simple nonlinear problems indicates the method of differentiation with respect to a parameter, Davidenko's method, when used in conjunction with perturbation analysis or a Newton-Raphson method, offers a powerful numerical technique for solving nonlinear problems. The main advantage of the continuation method is that it permits one to approach an exact solution* even when the initial estimate is not close to an exact solution. If one uses a method such as Newton-Raphson together with the continuation method, then one can also achieve good computational efficiency. The essence of the technique is to use the continuation method to obtain a rough first estimate and then use Newton-Raphson to refine the estimate. See Rheinboldt (1978).

Many other variations are possible. For example, Keller (1976, "Bifurcation Theory and Nonlinear Eigenvalue Problems," unpublished lecture notes, Caltech) poses a continuation problem in terms of an equation $G(\nu,\varepsilon) = 0$ where ν is

*This includes all local minima which satisfy the first order necessary conditions of optimality.

an element of a Banach space and $G(\cdot,\varepsilon)$ is an operator on that Banach space. Such an approach allows one to treat the TPBVP as a nonlinear operator equation on the Banach space of solution functions $x(\cdot,\varepsilon)$ of (3.8) - (3.10). Although seemingly more complicated than the approach of Kubicek-Hlavacek or Roberts-Shipman, this infinite dimensional point of view may offer the advantage of a numerically more stable solution algorithm.

The continuation methods seem very promising for application to the nonlinear TPBVP which appear in optimal control problems, and the technique deserves wider circulation.

Example

To see how continuation methods might be applied to aircraft trajectory optimization, consider the planar minimum-time-to-climb problem of Ardema (1976). Because of the absence of terminal constraints on horizontal position, the variables x and y are unnecessary. Also, the problem is solved in the vertical plane, so that the heading, β, and roll control variables are omitted. Thus, h, γ and V remain as state variables and velocity V is replaced by energy e. Also, thrust T is assumed constant, so that L (or α) is the only control variable.

This reduced system is of a dimension six which would allow a relatively inexpensive means of verifying and developing the various computational algorithms. The TPBVP for this system looks like

$$\dot{x} = f(x(t),n,\varepsilon)$$

where

$$x = (h,\gamma,e,\lambda_h,\lambda_\gamma,\lambda_E)$$

and

$$\dot{h} = V \sin\gamma$$

$$V\dot{\gamma} = L^* - \cos\gamma$$

$$\dot{e} = V(T - D)$$

$$\dot{\lambda}_h = \lambda_h(g/V)\sin\gamma - \lambda_\gamma(1/V^2)(g/V)(L^* - \cos\gamma) - \lambda_E \frac{\partial p}{\partial h}$$

$$\dot{\lambda}_\gamma = -\lambda_h V \cos \quad - \lambda_\gamma \sin\gamma/V$$

$$\dot{\lambda}_E = -\lambda_h(1/V) + \lambda_\gamma(1/V^2)(1/V)(L^* - \cos\gamma) - \lambda_E \frac{\partial p}{\partial E}$$

where

$$p = V(T - D)$$

$$D = D(L^{*2}, h, e)$$

and

$$L^* = \frac{\varepsilon\lambda_\gamma}{\lambda_E V^2 B}, \text{ the optimal control.}$$

In the expression L^*, $B = B(M)$ is an induced drag parameter, suitably dimensionalized for the available tabular data, and ε is the parameter used as the independent variable for solving the vector of unknown initial conditions, η, in the GPM method of Kubicek and Hlavacek (1972). In this example, $\eta = (\lambda_h(t_0), \lambda_\gamma(t_0), \lambda_e(t_0))$, the initial values of the three adjoint variables. The $\varepsilon = 0$ solution would be a starting point. (Note that this means $L^* = 0$.)

Other parameters are possible; for example, Breakwell (1977) uses $\varepsilon' = 1/(2(L/D)_{max})$ for the same problem. This is a coefficient of the induced drag term, so that the Hamiltonian would be linear in L for $\varepsilon' = 0$, producing a singular arc. Both formulations will be considered.

The boundary conditions are

$$h(t_0) = h_0, \quad \gamma(t_0) = \gamma_0(\varepsilon)$$

$$e(t_0) = e_0, \quad h(t_f) = t_f$$

$$e(t_f) = e_f, \quad \lambda_\gamma(t_f) = 0$$

The vector η defines the unspecified initial conditions.

This problem, then, is in a form to be solved by GPM, starting from $\varepsilon_0 = 0$ (or ε_0'). The algorithm involves integrating the parametric system

$$\frac{d\eta}{d\varepsilon} = -\Gamma^{-1}(\eta, \varepsilon)\frac{\partial F}{\partial \varepsilon}$$

where

$$F_i(x(t_f, \eta, \varepsilon)) = 0$$

is the $i^{\underline{th}}$ terminal boundary condition resulting from an integration of the

dynamic system to t_f from a given η, and

$$\Gamma_\eta(\eta,\varepsilon) = \text{the Jacobian } \{\partial F_i/\partial n_j\}$$

In this formulation the integration occurs for all ε in $[\varepsilon_0,\varepsilon_f]$, where ε_f is the desired value for ε. Note that, at each step in the integration, it is necessary to integrate the dynamic system to t_f, in general. (This may be relaxed to every $k^{\underline{th}}$ step of ε if convergence is good.) The differential equations arise from applying the implicit function theorem to the system $F_i = 0$.

Ultimately, it is desired to designate ε as the singular perturbation parameter. The application of schemes similar to GPM to this problem is not as straightforward, because in this role, $\varepsilon = 0$ reduces the order of the system. However, it is possible to transform the problem so that the scaled time parameter, $\tau \triangleq t/\varepsilon$, is the new independent variable in the dynamic system. The solution could then proceed from $\varepsilon = 0$ as follows: with $\varepsilon = 0$, compute the $0^{\underline{th}}$-order solution, including boundary layer solutions at both t_0 and t_f. Boundary layer matchings of solutions are necessary to assign values to system parameters to insure a stable integration. This $0^{\underline{th}}$-order solution for $x(t)$ could then be the first iteration in a continuation process, which would advance ε from 0 to a value small enough so that a first order linearized expansion is adequate.

We discuss continuation of singular perturbation problems further in Appendix 3.1. In addition, Appendix 3.1 discusses the problem of bifurcation in continuation. Appendix 3.2 describes a general continuation algorithm which we have implemented and which is based on the work of Kubicek (1976) and Keller (1977). Appendix 3.3 presents some simple numerical examples of TPBVP's solved by continuation. Finally, Appendix 3.4 contains a large bibliography of continuation method references.

APPENDIX 3.1
BIFURCATION AND SINGULAR PERTURBATION PHENOMENA IN CONTINUATION

The basic continuation problem is to solve the equation

$$G(x,\lambda) = 0 \tag{1.1}$$

for the vector x in terms of the real parameter λ. If B is a Banach space and G maps $B \times R$ into B, then we wish to find the trajectories $\lambda \to x(\lambda)$ in B which satisfy

$$G(x(\lambda),\lambda) = 0 \tag{1.2}$$

for all parameter values λ. For example, (1.1) might be the nonlinear equation for the missing initial values in the TPBVP. In this case, B would be a finite dimensional space, $B = R^n$. On the other hand, one might treat the entire trajectory control and state trajectory as a vector in an infinite dimensional vector space B. In that case, (1.1) would be a nonlinear operator equation on the Banach space B given by the Euler-Lagrange necessary conditions for the trajectory optimization problem. It appears that this infinite dimensional point of view offers the advantage of numerically more stable solution algorithms.

The method of differentiation with respect to a parameter first discussed by Davidenko (1953) solves the equation (1.1) for all values of the parameter λ by differentiating equation (1.2) with respect to λ to obtain

$$\frac{\partial G}{\partial x}(x(\lambda),\lambda)\,\frac{dx}{d\lambda} + \frac{\partial G}{\partial \lambda}(x(\lambda),\lambda) = 0 \tag{1.3}$$

Equation (1.1) is first solved at some value of the parameter λ, say $\lambda = \lambda_0$, and the solution $x(\lambda_0)$ is used as an initial condition from which to integrate the equation (1.3). The integration may proceed as long as the Frechet derivative $\frac{\partial G}{\partial x}(x(\lambda),\lambda)$, which is a linear operator from B into B for a given value of x and λ, is nonsingular and may be inverted to solve for the derivative $\frac{dx}{d\lambda}$ from (1.3). If $\frac{\partial G}{\partial x}(x(\lambda),\lambda)$ should become singular for some value of $\lambda = \lambda_1$, then a <u>bifurcation</u> has occurred at λ_1, indicating a sudden change in the nature and number of solutions of (1.1) in the neighborhood of λ_1.

In Figures 1 and 2 typical bifurcations* are illustrated. In Figure 1 there is one solution of (1.1) for values of λ near λ_1 such that $\lambda < \lambda_1$, and there are two solutions of (1.1) for values of λ near λ_1 such that $\lambda > \lambda_1$. In Figure 2 there is no solution of (1.1) for values of λ near λ_1 and such that $\lambda > \lambda_1$, but there are two solutions for values of λ near λ_1 such that $\lambda < \lambda_1$.

The proper treatment of bifurcations is extremely important in applying the continuation method to the solution of nonlinear problems for which there may be multiple solutions. Such a situation occurs, for example, if the Euler-Lagrange necessary conditions allow multiple extremal solutions of the trajectory optimization problem. One attractive feature of the continuation method is that it will yield all of the multiple solutions--provided that one has an algorithm that can handle bifurcations. We have implemented an algorithm due to Keller (1977) which handles both bifurcations pictured in Figures 1 and 2. For example, in Figure 1 the algorithm would trace out the left branch until λ approached λ_1; then it would indicate the singularity of $\frac{\partial G}{\partial x}(x(\lambda_1),\lambda_1)$ and proceed to trace the two right branches. In Figure 2 the algorithm would trace out the upper left branch until the parameter value reached λ_1. It would then indicate a singularity in $\frac{\partial G}{\partial x}(x(\lambda_1),\lambda_1)$ and reverse direction of the parameter λ to trace the lower left branch. This continuation algorithm is described in more detail in Appendix 3.2.

In solving a nonlinear equation $G(x) = 0$ by the continuation method, one first imbeds this problem in a one-parameter family of problems represented by (1.1) for which the parameter value $\lambda = 1$ gives the original problem and the parameter value $\lambda = 0$ gives a problem with a known solution. If one continues all of the solutions of (1.1) at $\lambda = 0$ to values of λ for which $\lambda > 0$, then one is guaranteed to find all the solutions of (1.1) at values of $\lambda > 0$, provided that one follows all the branches from bifurcations. In this way, one finds all the solutions of the original nonlinear problem $G(x) = 0$. In the case of an optimization problem, such as the trajectory optimization problem, one obtains all the extremal trajectories which satisfy the necessary conditions, and one may pick out the optimal solution from these extrema.

In addition to following all branches from bifurcations, it is also important to know all the initial solutions at $\lambda = 0$ and to continue each of these

*See note for Figure 2.

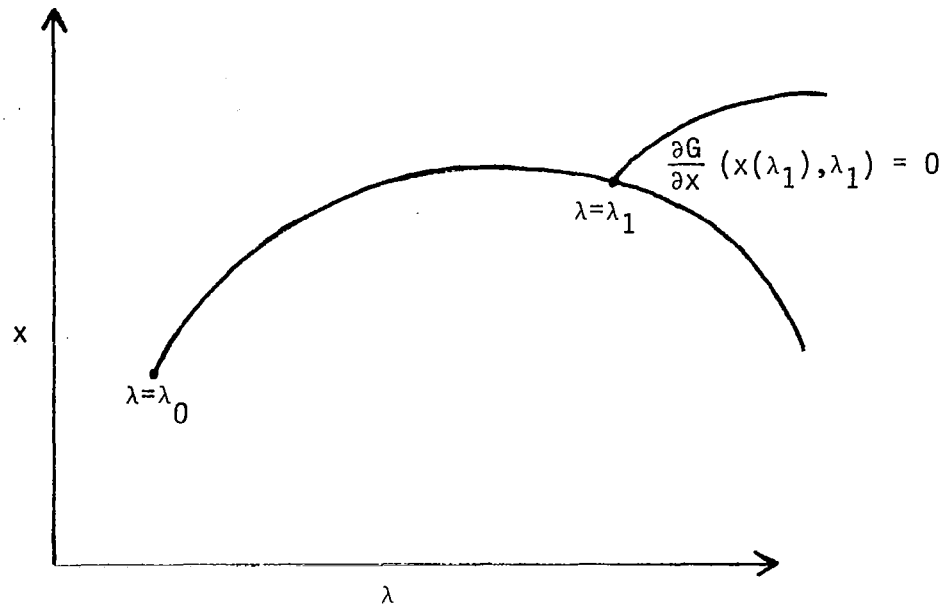


Figure 1.-Example of Bifurcation Point.

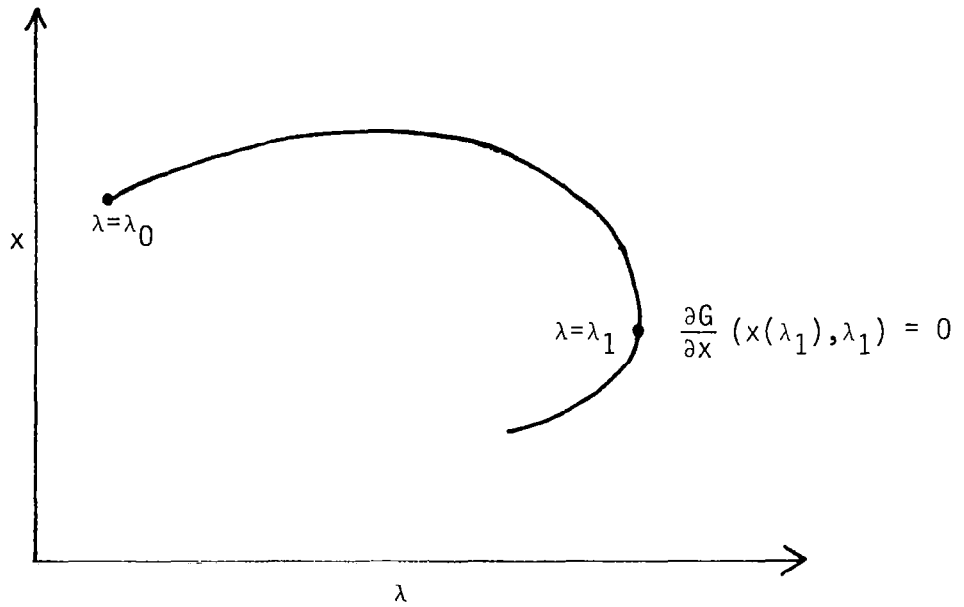


Figure 2.-Example of Limit Point.

NOTE: Unhappily, as in the case of the word "singular," "bifurcation" has come to have at least two distinct meanings. As opposed to the above definition, several authors, e.g. Keller (1977), designate only the λ_1 in Fig. 1 as a bifurcation point, making it a subset of our definition. In this case, λ_1 in Fig. 2 is named a limit point. The cases of Fig. 1 and Fig. 2 are distinguishable mathematically, as will be shown in Appendix 3.2.

solutions to values of $\lambda > 0$. One usually selects the one-parameter imbedding (1.1) so that it yields only one solution $x(\lambda)$ for λ in a neighborhood of 0. However, it sometimes happens that (1.1) has no solution at $\lambda = 0$ although it has a unique asymptotic behavior as $\lambda \to 0$. Such is the situation in problems such as our aircraft trajectory optimization problem, in which the parameter λ represents a singular perturbation near 0. For example, the parameter λ is denoted by ε in the aircraft optimization problem represented by (4.2.46), (4.2.47) in Section 4.2. These equations have no solution at $\lambda = 0$ since it is generally impossible to satisfy all the boundary conditions for the reduced order ($\lambda = 0$) equations. However, the optimal solution has a well-defined, unique asymptotic behavior as $\lambda \to 0$. In this case, it is still possible to apply the continuation method successfully by starting the process at a $\lambda_0 > 0$ in a neighborhood of $\lambda = 0$ and by using the asymptotic approximation of the solution $x(\lambda_0)$ as the initial condition in the differential equation (1.3). To do this, one may have to make preliminary Newton corrections to obtain a more exact initial condition at λ_0, but for small nonzero values of the parameter λ the asymptotic approximation is very accurate and only a few Newton corrections are usually necessary. The implemented continuation algorithms feature such preliminary Newton correction, as well as Newton correction after every prediction step, to improve accuracy to a pre-specified tolerance.

Continuation from a singular perturbation will be effective when the asymptotic approximation is good for λ near $\lambda = 0$ but not very good for λ near the desired solution at $\lambda = 1$. In particular, this technique will improve the asymptotic approximation to the aircraft optimization problem when the total range for interception decreases below the lower limit for validity of the energy state approximation.*

*This lower limit is about 160-170 Km for the F-4 example problem.

APPENDIX 3.2
GENERAL CONTINUATION ALGORITHMS

A series of numerical algorithms for the solution of nonlinear systems of the form (A3.1.1) has been developed for application to such problems as aircraft trajectory optimization. Ultimately, it is hoped to combine these algorithms into one system. In this way, the simplest (and presumably the most time/core-efficient) algorithm may be selected initially, switching to more complete algorithms as the numerical development of the problem warrants. The basis for the developed algorithms lies principally in work by Klopfenstein (1961), Kubicek (1973, 1976), Keller (1977) and Rheinboldt (1977).

Each of the aforementioned researchers realized that effective implementation of the parametric system of Davidenko (A3.1.3) requires adequate treatment of the Frechet derivative

$$(2.1) \qquad F(x(\lambda),\lambda) = \frac{\partial G}{\partial x}(x(\lambda),\lambda)$$

when λ is in the neighborhood of some point λ_1 at which F becomes singular. F is singular at either point λ_1 in Figure 1 of Appendix 3.1 (a "proper" bifurcation point) or Figure 2 (a "limit" point). Rheinboldt and Keller have developed a means of extending the continuation through the Figure 1-type bifurcation point. This point differs from the limit point of Figure 2 in a mathematical sense, as will be seen below. Limit points are somewhat easier to deal with, and so will be discussed first.

As can be seen in Figure 2 (especially if one imagines x and G to be scalar), $dx/d\lambda \to \infty$ as $\lambda \to \lambda_1$ from λ_0. Hence, F must be singular at λ_1 for the trajectory to be meaningful. It is also precisely at λ_1 where the parameter λ loses its monotonic property. Because the connection is not coincidental, it seemed reasonable to augment the problem by introducing an arclength parameter, say t, which is by nature monotonic. The problem is augmented in that the system parameter λ is now itself a dependent variable in the arclength parameter. Thus, if $x \in R^n$, the augmented system is based on the solution of

$$(2.2) \qquad G(x(t),\lambda(t)) = 0$$

or

(2.3) $G(y(t)) = 0$

where

(2.4) $y = (x,\lambda), \quad y \in R^{n+1}$

Since there are now n+1 unknowns, (2.2) must be augmented by the particular arclength relationship. Both Klopfenstein and Kubicek use a purely Euclidean relationship (called a "normalization")

(2.5) $N_1(x,\lambda,t) = 0 = \dot{x}_1^2 + \ldots + \dot{x}_n^2 + \dot{\lambda}^2 - 1$

where

(2.6) $(\dot{}) \triangleq \frac{d(\ )}{dt}$

The system can now be solved, as follows:

(2.7) $\frac{dG}{dt} \triangleq \dot{G} = \frac{\partial G}{\partial x}\dot{x} + \frac{\partial G}{\partial \lambda}\dot{\lambda} = 0$

Equation (2.7) is the starting point for both methods of Klopfenstein and Kubicek. It may be rewritten

(2.8) $A\dot{y} = 0$

where

(2.9) $A = \left[\frac{\partial G}{\partial x}, \frac{\partial G}{\partial \lambda}\right] = \left[F, \frac{\partial G}{\partial \lambda}\right]$

Equation (2.8) is a system of n equations for the n+1 unknown elements of y. Klopfenstein does not distinguish between the x_i and λ, considers $\lambda = x_{n+1}$. His method depends on finding an x_{n+1} for which F remains non-singular. Then,

(2.10) $\dot{x} = -F^{-1}\left(\frac{\partial G}{\partial \lambda}\right)\dot{\lambda} = -F^{-1}\left(\frac{\partial G}{\partial x}\right)\dot{x}_{n+1}$

Equation (2.5) then determines $\dot{x}_{n+1}$.

The method of Kubicek is more "robust" in that, while he also considers λ

and x functionally equivalent in terms of t, the most non-singular $n \times n$ submatrix of A is used at each point for the matrix inversion operation. This is achieved by Gaussian elimination with controlled pivoting. By this procedure, one of the n+1 columns of A for which A is "most singular" is eliminated, say column k. Then $x_k \leftrightarrow y_k$ plays the role of the parameter λ, instead of always using y_{n+1}. Equation (2.7) is then rearranged and solved for the n $\dot{y}_i$, $i \neq k$:

$$\frac{dG}{dt} = F_k[\dot{y}_i]_{i\neq k} + \frac{\partial G}{\partial y_k}\dot{y}_k = 0 \tag{2.11}$$

where

$$F_k = \begin{bmatrix} \frac{\partial G_1}{\partial y_1}, \cdots, \frac{\partial G_1}{\partial y_{k-1}}, \frac{\partial G_1}{\partial y_{k+1}}, \cdots, \frac{\partial G_1}{\partial y_{n+1}} \\ \vdots \\ \frac{\partial G_i}{\partial y_1} \\ \vdots \\ \frac{\partial G_n}{\partial y_1}, \cdots, \frac{\partial G_n}{\partial y_{k-1}}, \frac{\partial G_n}{\partial y_{k+1}}, \cdots, \frac{\partial G_n}{\partial y_{n+1}} \end{bmatrix} \tag{2.12}$$

$$\dot{y}_i = -F_k^{-1}\left(\frac{\partial G}{\partial y_k}\right)\dot{y}_k, \quad i \neq k \tag{2.13}$$

Note that F_k is $n \times n$, a square matrix. Equation (2.13) is similar in form to (2.10), but more general.

As before, the $\dot{y}_i$ are substituted into (2.5) to solve for the final element, $\dot{y}_k$. The sign ambiguity is handled by selecting a sign at the starting point, y_0, $t = 0$, which is consistent with the problem, realizing that t increases monotonically. For example, if rudder deflection, δr, is selected as y_k and has the value $+\delta r_{max}$ at the starting point, then the negative sign on $\dot{y}_k$ would be selected, so that continuation may proceed into the acceptable range of δr values.

The Gaussian elimination procedure works as follows: all of the elements of A are scanned for the one with the largest magnitude. This element becomes the pivot point for the elementary matrix row operations which are used to zero

all remaining elements in the column of the pivot element. This column is saved and the scanning process begins again, for the remaining columns. This process is only done n times, leaving untouched the column whose elements are consistently the smallest. The untouched column becomes column k. The remaining n columns can be rearranged to produce a diagonal matrix, whereupon inversion is straightforward.

Again, the method works well at limit points because, say at λ_1 in Figure 2 of Appendix 3.1, λ would cease to be the parameter, but would be replaced by one of the x_i. It should also be mentioned that provision can be made for influencing the choice of y_k by scaling each of the columns of A by a scalar, thus reducing their magnitudes.

Given that the $n+1$ $\dot{y}_i$ have somehow been found at a certain point $t \in [0,1]$, the continuation process of Kubicek evolves essentially by integrating the system (2.13) and (2.5) for $y(t)$. The continuation proceeds numerically by a predictor-corrector sequence. At the starting point ($t = 0$, $\lambda = \lambda_0$), the x_i are found to the required precision using Newton-Raphson. Then, Adams-Bashforth variable order (≤ 4) is used to advance all of the y_i--i.e., x_i and λ--to a predicted value. At this new point, y_k is found as described above, and Newton corrections are made on the y_i, $i \neq k$, until reasonable convergence is assured. Typically, no more that three or four Newton-corrector steps are needed at each point. The Newton-corrector formula is

$$y^{(j+1)} = y^{(j)} - F_k^{-1} G\left(y^{(j)}, y_k\right) \tag{2.14}$$

where $y^{(\cdot)}$ is an $n \times 1$ vector of all y_i, $i \neq k$.

The Klopfenstein/Kubicek algorithm was also tested on another project. In this application, equilibrium solutions of an aircraft are generated in both developed spin and roll departure flight regimes. The continuation system is the equations of motion, with the derivatives set to zero. One of the three aerosurface controls is chosen as the nominal continuation parameter, and the other two are set to fixed values. For spin analysis, an eighth order system of equations is needed, because of coupling effects arising from the need to employ the full nonlinear expressions. (The roll departure regime requires only a fifth order system, but is still highly nonlinear.) However, the continuation algorithm described above is able to solve for the equilibrium surfaces

quite readily. This algorithm is also the basis for some of the two-point boundary value problem (TPBVP) continuation examples which are described in Appendix 3.3. For TPBVP applications, the system (2.5), (2.13) operates in function space rather than Euclidean n-space. The significant difference is that, for every continuation step ("outer" integration), a full solution must be generated between the two boundary points by quadrature ("inner" integration). While the function-space process is thus considerably more complicated, the continuation itself is non-iterative; that is, the continuation parameter moves directly from its initial to its final value. There is no retracing or repetition of this outer loop quadrature.

For flight trajectory optimization problems, the TPBVP system consists of adjoint, or influence, functions whose initial conditions are unknown in general. These, then, become the variables x in (2.2) and λ, the continuation parameter, is usually chosen to be a physical parameter such as $(L/D)_{max}$ or air density. This choice enables a reasonably simple solution to be found at $\lambda = 0$. See Appendix 3.3 for a discussion of continuation methods applied to the solution of TPBVP's.

At this point, it is probably easy to appreciate that it is quite often convenient to evaluate the elements of A numerically, rather than to pre-compute analytic expressions. This is particularly the case when simulating flight vehicle trajectories, because such expressions involve terms containing tabular data. Analytic expressions for such terms are typically very complicated. Therefore, a numerical differentiation algorithm has been developed and is now a part of our basic continuation algorithm. It may be invoked at the user's option.

The algorithm which thus far has worked most efficiently is the following: to compute $F \triangleq \frac{\partial G}{\partial x}$ at $x = x_0$, fit a cubic spline to G(x) at the following five points (knots):

$$G(x + i\Delta), \quad i = 0, \pm 1, \pm 2 \tag{2.15}$$

Once this fit is made, the slope of the spline function at x_0 becomes the approximation to $F(x_0) \triangleq F_0$. In general, G may be a vector and x a scalar. If $x \in R^n$, the extension is obvious; one merely performs the operation n times for each x_j.

It is important to use a value for Δ which is small enough to adequately represent F_0, yet not so small that numerical difficulties result. Thus, we require*

$$(2.16) \qquad |G(x) - G(x+\Delta)| \simeq O(\varepsilon)$$

where $\varepsilon = 0.0001$ has been shown to produce good results.

The algorithm for selecting Δ is as follows:

1) initialize $i = 1$
2) set $\Delta_i = \varepsilon$ (this is exact if $F(x_0) = 1$)
3) compute $G_i = |G(x_0) - G(x_0 + \Delta_i)|$
4) if $\frac{\varepsilon}{10} < G_n < 10\varepsilon$, go to 7
5) if not, set $\Delta_{i+1} = \Delta_i \left(\frac{\varepsilon}{G_i}\right)$

   Note that $G_i = 0$ where the slope is infinite
6) set $i = i+1$ and go to 3
7) set $\Delta = \Delta_i$ and do the spline fit

If a solution for Δ is not obtained after five iterations, a warning is printed and Δ is set to Δ_5.

As mentioned above, the Kubicek algorithm is unable to solve automatically for all of the branches which emanate from the bifurcation point λ_1 shown in Figure 1 of Appendix 3.1. We have implemented, therefore, a method based on the work of Rheinboldt (1977) and Keller (1977) which can not only continue the original branch accurately past λ_1, but also accurately evaluate λ_1 and the slope of the "secondary" branch at λ_1. With this slope, continuation along the secondary branch can proceed as usual.

Keller's algorithm begins with the system (2.2) augmented by the "pseudo-arclength" normalization

$$(2.18) \qquad N_3 \triangleq \theta \dot{x}_0^T (x - x_0) + (1-\theta)\dot{\lambda}(\lambda - \lambda_0) - (t - t_0) = 0$$

where t is the arclength parameter, θ is a constant selected such that $0 < \theta < 1$, and $(x_0, \lambda_0) = y_0$ are the values of the unknowns and the parameter at t_0. The

---

*If $G \in R^n$, (2.16) is modified:

$$(2.17) \qquad \frac{1}{n} \sum_{i=1}^{n} |G_i(x) - G_i(x+\Delta)| \simeq O(\varepsilon)$$

augmented system is now:

$$(2.19) \qquad H(y,t) \triangleq \begin{bmatrix} G(y) \\ N_3(y,t) \end{bmatrix} = 0$$

where

$$(2.20) \qquad y = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \quad y \in R^{n+1}$$

Keller has shown that, using N_3 as given by (2.18), the quantity

$$(2.21) \qquad B \triangleq \frac{\partial H}{\partial y}$$

is nonsingular if and only if

a) $F = \frac{\partial G}{\partial x}$ is nonsingular; or

b) $\frac{\partial G}{\partial \lambda} \notin R(F)$, where $R(\cdot)$ denotes range space.

Case (b) corresponds to a limit point. At such a point (e.g., Figure 2 of Appendix 3.1) there is no intersection of branches, but $dx/d\lambda \to \infty$. However, solution of the augmented system (2.19) continues normally.

If neither condition (a) or (b) holds at some point λ_1, then B is singular and λ_1 is the type of bifurcation point shown in Figure 1, where two or more branches intersect. As mentioned above, λ_1 is skipped over, in order to continue along the initial branch. Any predictor-corrector method will suffice for this. Continuing the solution along the second branch then proceeds as follows*:

1) compute $\partial G/\partial x$ (singular) at the bifurcation point, and get an approximation for dx/dt and $d\lambda/dt$.

2) compute the eigenvalues and eigenvectors of $\partial G/\partial x$. Call the eigenvector corresponding to the zero eigenvalue ϕ_0, and the others $\phi_1, \ldots, \phi_{N-1}$.

3) compute ϕ_N in the following:

$$(2.22) \qquad (\partial G/\partial x)\phi_N + \partial G/\partial \lambda = 0$$

where ϕ_N is a linear combination of the vectors $\phi_1, \ldots, \phi_{N-1}$. The vectors ϕ_0 and ϕ_N define the plane in which a solution for $\partial x/\partial t$ can appear.

4) solve for the coefficients τ_0 and τ_N in:

---

*This is Keller's (1977) "Method II."

(2.23) $$\hat{\tau}_0 = \tau_N(1 + \|\phi_N\|^2)$$

$$\hat{\tau}_N = -\tau_0 \|\phi_0\|^2$$

6) now we seek a solution of the form

(2.25) $$x^1 = x_0 + \varepsilon[\hat{\tau}_0\phi_0 + \hat{\tau}_N\phi_N] + V$$

$$\lambda^1 = \lambda_0 + \varepsilon\hat{\tau}_N + \eta$$

where x_0, λ_0 is the solution at the bifurcation point.

7) finally, starting with $V = 0$, $\eta = 0$, use Newton's method to find a solution to the equations:

(2.26a) $$G(x^1, \lambda^1) = 0$$

(2.26b) $$(\hat{\tau}_0\phi_0^* + \hat{\tau}_N\phi_N^*)V + \hat{\tau}_N\eta = 0$$

The ε terms in (2.25) move the possible solution away from the first branch, and equation (2.26b) insures that the free variables V and η do not bring the solution back. Thus, any solution found must be on the second branch of the solution curve, and can be used as a starting point for a normal continuation solution.

This method has been implemented and run on test problems. Because of its complexity, especially in the use of TPBVP continuation algorithms, much effort is required to select numerical parameters and quadrature techniques which will efficiently deal with the system at hand.

APPENDIX 3.3

APPLICATION OF CONTINUATION ALGORITHM TO TPBVP: NUMERICAL EXAMPLES

The algorithm described in Appendix 3.2 was applied to solving three example TPBVP's by the shooting method. The theory behind the shooting method is presented in Subsection 3.3.3 and we refer the reader there for details of the basic method. Note that the present method of solving the TPBVP by shooting uses both the Davidenko method of differentiation with respect to a parameter described by Kubicek and Hlavacek (1972a, 1972b, 1973) and Kubicek (1976) and the Newton-Raphson method described by Roberts and Shipman (1967, 1968) together in one algorithm.

Example #1

This is a nonlinear second order two-point boundary value problem which we solved by continuation with respect to the final time as described in Roberts and Shipman (1967, 1968, 1972). The problem was to solve

$$\begin{aligned} \frac{du_1}{dt} &= u_1 \ln u_2 \\ \frac{du_2}{dt} &= u_2 \ln u_1 \\ u_1(0) &= 1, \quad u_2(T) = e \end{aligned} \tag{3.1}$$

for several given values of the final time T. The problem was solved numerically by continuation with respect to the final time T, starting at $T = 0$ where the problem reduces to an initial value problem with $u_1(0) = 1$ and $u_2(0) = e$. The problem in (3.1) was chosen to be the nonlinear transformation of the linear problem

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= x_1 \\ x_1(0) &= 0, \quad x_2(T) = 1, \end{aligned} \tag{3.2}$$

via the transformation $u_1 = e^{x_1}$ and $u_2 = e^{x_2}$ so that we would have an exact solution to the problem (3.1). If b(T) denotes the missing initial condition $u_2(0) = b(T)$ in (3.1) for the problem with final time T given, then the function b is given by the formula

$$b(T) = \exp(2/(e^{T} + e^{-T})) \tag{3.3}$$

The numerical continuation solution agreed with (3.3) exactly in all displayed decimal digits.

Example #2

This is also a nonlinear, second order TPBVP but the final time is fixed and an internal parameter is used for continuation. This example represents the equilibrium equation for the heat distribution in a rod of length x = 1. The problem is solved by continuation in Wasserstrom (1973) which is an excellent review of continuation methods in general. The equations for the problem are

$$\begin{aligned} \frac{dy_1}{dx} &= y_2 \\ \frac{dy_2}{dx} &= \lambda \exp(y_1) \\ y_1(0) &= y_2(1) = 0 \end{aligned} \tag{3.4}$$

where λ is the continuation parameter used. We applied continuation to the shooting method to calculate numerically the missing initial condition $y_2(0) = b(\lambda)$ for values of the parameter λ between 0 and 1. Note that for $\lambda = 0$ the problem has a trivial solution, $y(x) = 0$ for all x.

The exact solution of (3.4) is given by the formula

$$y(x) = \ln([m^2/2\lambda]\sec^2(\tfrac{m}{2}[x - \tfrac{1}{2}])) \tag{3.5}$$

where m must satisfy the implicit equation

$$\frac{m^2}{2\lambda} \sec^2 \frac{m}{4} = 1 \tag{3.6}$$

The missing initial condition $b(\lambda)$ is given in terms of m and λ as

$$(3.7) \qquad b(\lambda) = -m \tan(n/4) = -\sqrt{2\lambda - m^2}$$

Again, the numerical continuation solution agreed exactly with (3.7) in all displayed decimal digits.

Note that our continuation algorithm used numerical differentiation to calculate the partial derivatives necessary for the shooting method instead of integrating four extra variational differential equations as Wasserstrom does in his paper. The numerical differentiation simplifies the programming requirements and will reduce the computation required in large order problems.

Example #3

The last example is a trajectory optimization problem for a simplified missile interceptor taken from the paper of Schneider and Reddy (1974). Although the aerodynamic model is extremely simplified, this problem is a good example with which to test out the continuation algorithm before attempting the aircraft equations. The state equations in this problem are nonlinear and fourth order, resulting in a nonlinear, eighth order TPBVP. The nonlinear, fourth order state equations are

$$(3.8) \qquad \begin{aligned} \frac{dx_1}{dt} &= x_3 \\ \frac{dx_2}{dt} &= x_4 \\ \frac{dx_3}{dt} &= -\varepsilon x_3 v e^{-x_2/h_s} + u_1 \\ \frac{dx_4}{dt} &= -\varepsilon x_4 v e^{-x_2/h_s} + u_2 - 9.81 \end{aligned}$$

where $v = \sqrt{(x_3^2 + x_4^2)}$.

In the equations (3.8), x_1 represents a horizontal range variable and x_2 is the height variable. The variables x_3 and x_4 are the velocities corresponding to x_1 and x_2 respectively. The variables u_1 and u_2 are the controls (thrusts) for the problem. The parameter h_s is a height scale which is taken as 6705.6 m in this problem. The parameter ε is equal to $\frac{b}{2}\rho_s$ where ρ_s is the sea level air density and b is the inverse ballistic coefficient for the missile

interceptor modeled by (3.8). The optimality criterion for this problem is the simplest quadratic criterion,

$$(3.9) \qquad \int_0^T (u_1^2 + u_2^2)dt$$

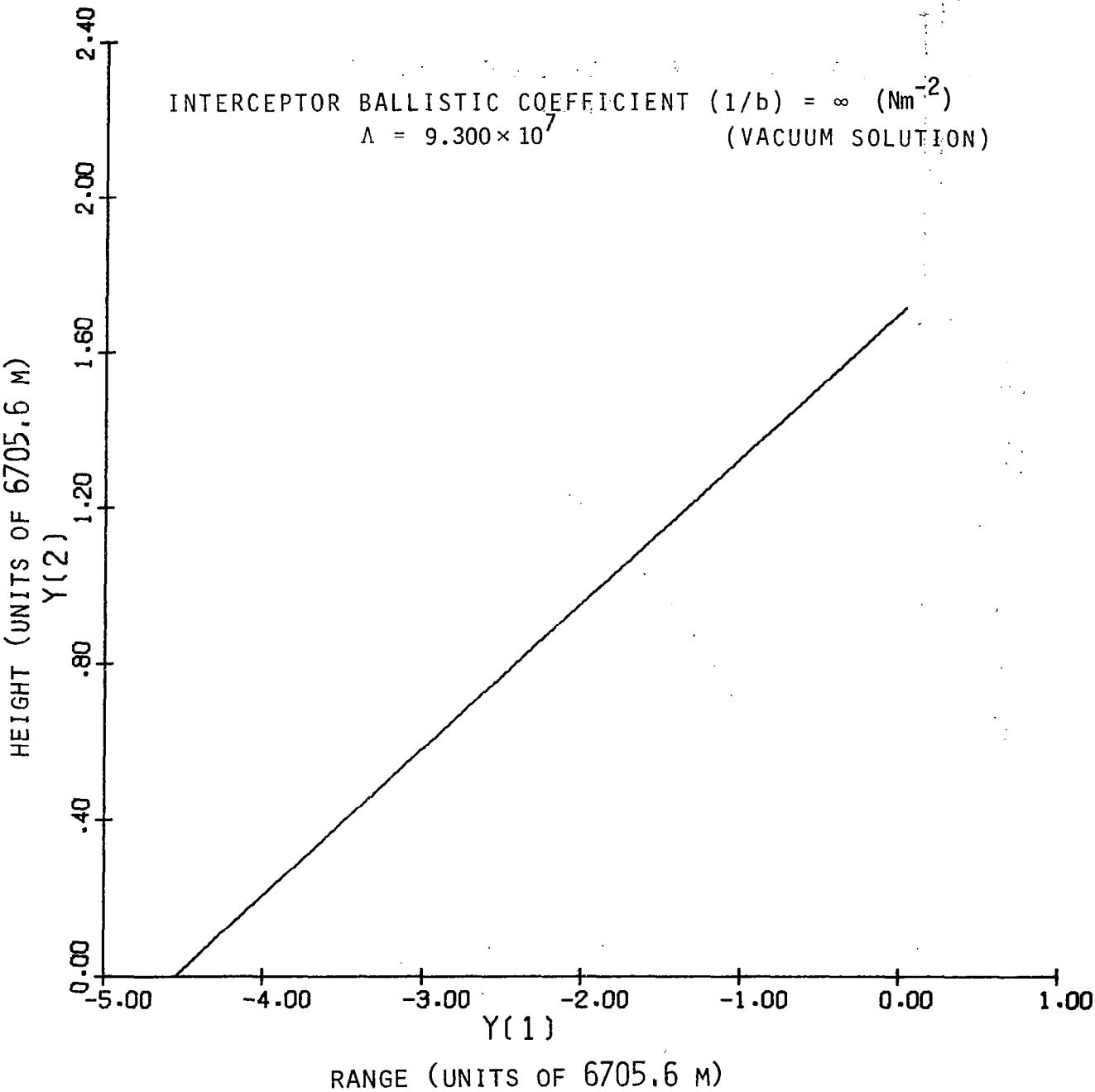
where T is the fixed final time. The Euler-Lagrange necessary conditions give the following nonlinear, eighth order TPBVP:

$$\begin{aligned}
\frac{dx_1}{dt} &= x_3 \\
\frac{dx_2}{dt} &= x_4 \\
\frac{dx_3}{dt} &= -\varepsilon x_3 v e^{-x_2/h_s} - \frac{x_7}{2} \\
\frac{dx_4}{dt} &= -\varepsilon x_4 v e^{-x_2/h_s} - \frac{x_8}{2} - 9.81 \\
\frac{dx_5}{dt} &= 0 \\
\frac{dx_6}{dt} &= -\frac{\varepsilon v}{h_s} e^{-x_2/h_s}\left[x_7 x_3 + x_8 x_4\right] \\
\frac{dx_7}{dt} &= -x_5 + \varepsilon x_7\left[1 + \left(\frac{x_3}{v}\right)^2\right] v e^{-x_2/h_s} + \varepsilon x_8\left[\frac{x_4 x_3}{v^2}\right] v e^{-x_2/h_s} \\
\frac{dx_8}{dt} &= -x_6 + \varepsilon x_7\left[\frac{x_4 x_3}{v^2}\right] v e^{-x_2/h_s} + \varepsilon x_8\left[1 + \left(\frac{x_4}{v}\right)^2\right] v e^{-x_2/h_s}
\end{aligned} \qquad (3.10)$$

where $x_1(0)$, $x_2(0)$, $x_3(0)$, $x_4(0)$, $x_1(T)$, $x_2(T)$, are given and $x_7(T) = x_8(T) = 0$. As in (3.8), $v = \sqrt{(x_3^2 + x_4^2)}$. In equation (3.10), the variables x_5, x_6, x_7 and x_8 are the adjoint variables corresponding to x_1, x_2, x_3 and x_4 respectively. We solved (3.10) by shooting for the missing adjoint initial conditions $x_5(0)$, $x_6(0)$, $x_7(0)$ and $x_8(0)$. The parameter ε (proportional to the inverse ballistic coefficient) was used for continuation from $\varepsilon = 0$. Note that the $\varepsilon = 0$ solution

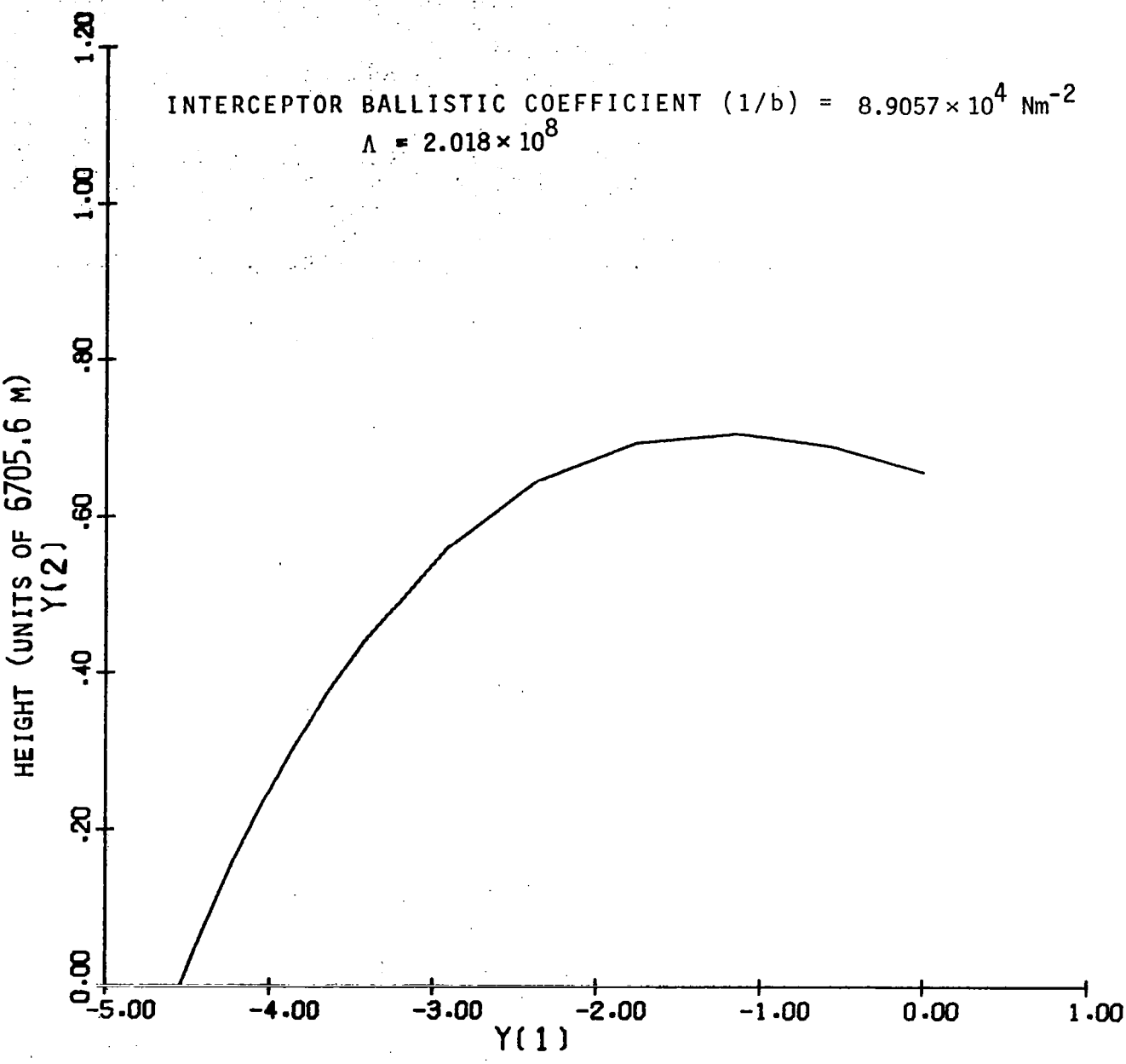
corresponds to an interception problem with no air density--i.e., a vacuum solution. Figures 1, 2, 3 and 4 show the range-height trajectories for four values of the ballistic coefficient ranging from ∞ to 3.1266×10^{4} Nm^{-2}. This corresponds to continuing the parameter ε from 0 (the vacuum solution) to 1.9220×10^{-4} m^{-1}. The final time T is 20 sec and the given initial and final conditions are $x_2(0) = x_3(0) = x_4(0) = 0$, $x_1(0) = -1.524 \times 10^{4}$ m, $x_1(T) = 0$, $x_2(T) = 4.409 \times 10^{3}$ m. These initial and final conditions correspond to the conditions for cases 5 and 6 in Schneider and Reddy (1974). Schneider and Reddy presented the trajectory of case 6 which corresponds to our Figure 4, although the ballistic coefficient for case 6 is about two times that of Figure 4. Figure 3 is very close to case 5, although the ballistic coefficient for Figure 3 is 4.6923×10^{4} Nm^{-2} which is slightly less than the coefficient for case 5, namely 5.788×10^{4} Nm^{-2}. The cost associated with Figure 3 ($\Lambda = 3.542 \times 10^{8}$) is correspondingly slightly larger than the cost in case 5 ($\Lambda = 3.415 \times 10^{8}$).

As the parameter ε and the final time T increase, the equations (3.10) become more sensitive to slight changes in the initial values of the adjoint variables. This is partly due to the increased nonlinearity of the problem, but the main trouble comes from the forward integration of the adjoint equations. The adjoint equations are unstable in the forward direction and the differential equations (3.10) will become infinite in a finite amount of time. One solution to the problem which still maintains the shooting method is to use double precision instead of single precision accuracy in computations. Alternatively, one can use one of the function space methods such as quasilinearization as presented in Roberts and Shipman (1968a, 1972) or the back-and-forth shooting method of Orava and Lautala (1976, 1977).



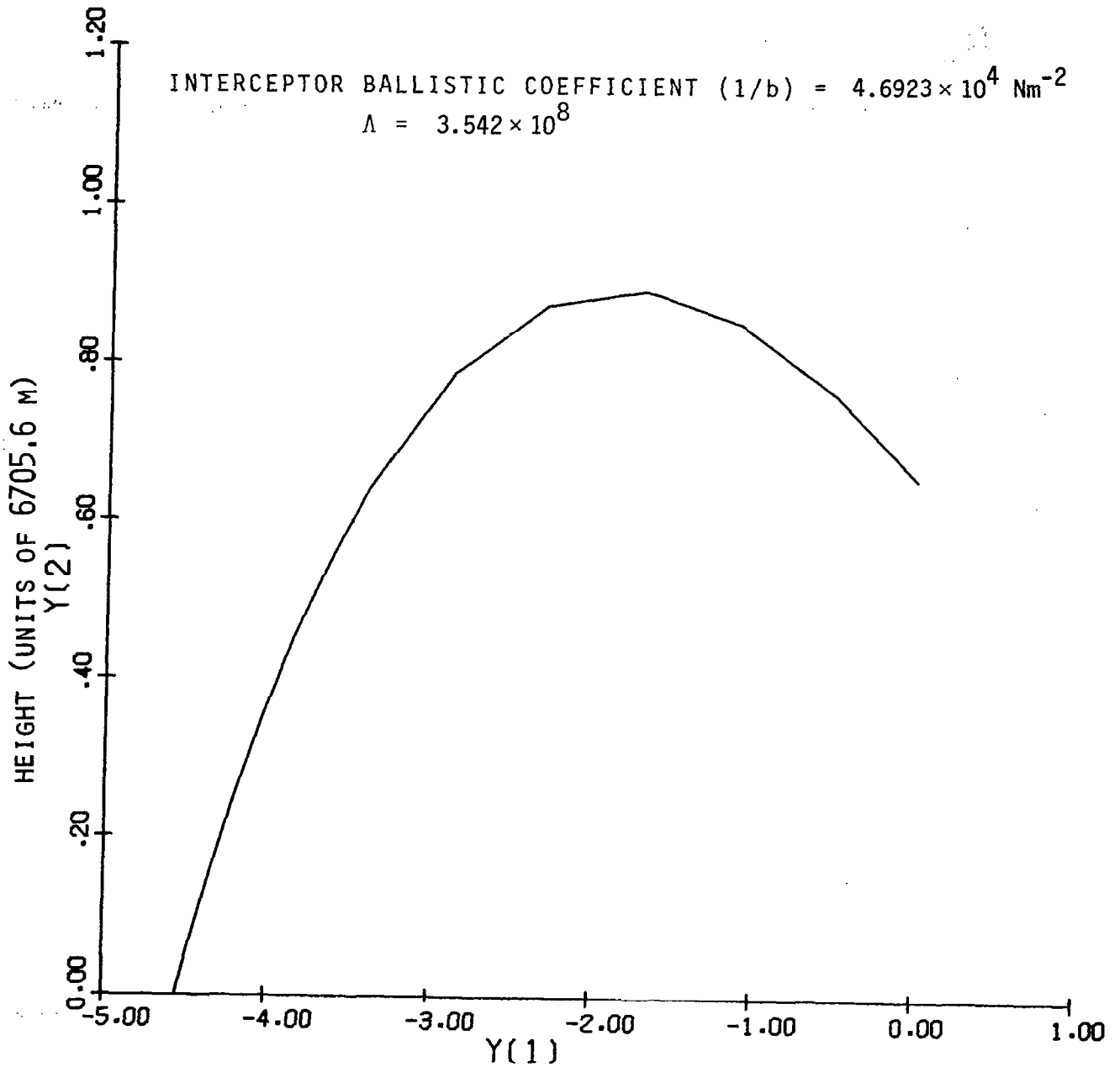
SCHNEIDER-REDDY EXAMPLE

Figure 1



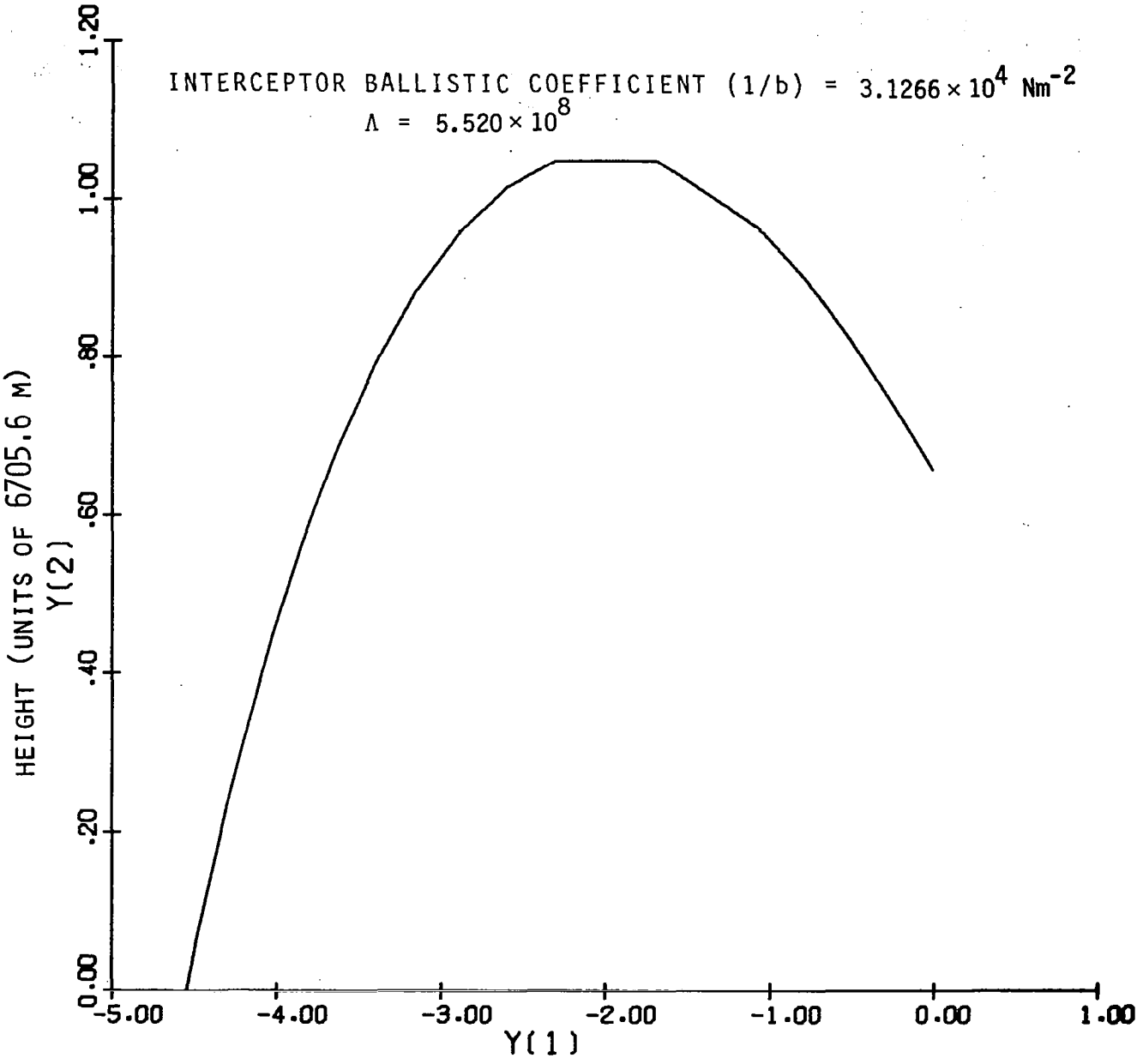
SCHNEIDER-REDDY EXAMPLE

Figure 2



SCHNEIDER-REDDY EXAMPLE

Figure 3



SCHNEIDER-REDDY EXAMPLE

Figure 4

APPENDIX 3.4

CONTINUATION METHOD BIBLIOGRAPHY

1. Anselone, P. M. and R. H. Moore (1966), "An Extension of the Newton-Kantorovic Method for Solving Nonlinear Equations with an Application to Elasticity," J. Math. Anal. Appl., Vol. 13, pp. 476-501.

2. Avila, J. (1970), "Continuation Methods for Nonlinear Equations," PhD thesis, Univ. of Maryland, College Park, MD.

3. Bauer, L., H. B. Keller and E. L. Reiss (1970), "Axisymmetric Buckling of Hollow Spheres and Hemispheres," Comm. Pure Appl. Math., Vol. 23, pp. 529-568.

4. Boggs, P. T. (1971), "The Solution of Nonlinear Systems of Equations by A-Stable Integration Techniques," SIAM J. Numer. Anal., Vol. 8, pp. 767-785.

5. Bosarge, W. (1971), "Iterative Continuation and the Solution of Nonlinear Two-Point Boundary Value Problems," Numer. Math., Vol. 17, pp. 268-283.

6. Brayton, R. K., F. G. Gustavson and G. D. Hachtel (1972), "A New Efficient Algorithm for Solving Differential-Algebraic Systems Using Implicit Backward Differentiation Formulas," Proc. IEEE, Vol. 60, pp. 98-108.

7. Broyden, D. G. (1969), "A New Method of Solving Nonlinear Simultaneous Equations," Comput. J., Vol. 12, pp. 94-99.

8. Casti, J., R. Kalaba, G. Meier and E. Zagustin (1974), "Interval Length Imbedding for Nonlinear Two-Point Boundary Value Problems," JOTA, Vol. 13, pp. 159-163.

9. D'Ans, G., Y. Hontour and M. Jamshidi (1970), A Manifold Imbedding Solution of Optimal Control Problems," Proc. 8th Allerton Conf., Monticello, IL, pp. 564-575.

10. Davidenko, D. (1953a), "On a New Method of Numerically Integrating a System of Nonlinear Equations," Dokl. Akad. Nauk. SSSR, Vol. 88, pp. 601-604.

11. Davidenko, D. (1953b), "On the Approximate Solution of a System of Nonlinear Equations," Ukrain. Math. Z., Vol. 5, pp. 196-206.

12. Davidenko, D. (1955), "On the Application of the Method of Variation of Parameters to the Theory of Nonlinear Functional Equations," Ukrain. Math. Z., Vol. 7, pp. 18-28.

13. Davidenko, D. (1965a), "An Application of the Method of Variation of Parameters to the Construction of Iterative Formulas of Increased Accuracy for Numerical Solutions of Nonlinear Integral Equations," Dokl. Akad. Nauk. SSSR, Vol. 162, pp. 499-502, and Soviet Math. Dokl., Vol. 6, pp. 702-706.

14. Davidenko, D. (1965b), "An Application of the Method of Variation of Parameters to the Construction of Iterative Formulas of Higher Accuracy for the Determination of the Elements of the Inverse Matrix," Dokl. Akad. Nauk. SSSR, Vol. 162, pp. 743-746, and Soviet Math. Dokl., Vol. 6, pp. 738-742.

15. Davis, J. (1966), "The Solution of Nonlinear Equations with Critical Points," PhD thesis, Oregon State Univ., Corvallis, OR.

16. Deist, F. and L. Sefor (1967), "Solution of Systems of Nonlinear Equations by Parameter Variation," Comput. J., Vol. 10, pp. 78-82.

17. Eaves, B. C. and R. Saigal (1972), "Homotopies for the Computation of Fixed Points on Unbounded Regions," Math. Prog., Vol. 3, pp. 225-237.

18. Ehrman, H. (1963), "On Implicit Function Theorems and the Existence of Solutions of Nonlinear Equations," Enseignement Math., Vol. 9, pp. 129-176.

19. Ficken, F. (1951), "The Continuation Method for Functional Equations," Comm. Pure Appl. Math., Vol. 4, pp. 435-456.

20. Freudenstein, F. and B. Roth (1963), "Numerical Solution of Systems of Nonlinear Equations," J. Assoc. Comput. Math., Vol. 10, pp. 550-556.

21. Glasser, D. (1969), "Numerical Solution of Two-Point Boundary Value Problems on Total Differential Equations," SIAM J. Numer. Anal., Vol. 6, pp. 591-597.

22. Glasser, D. and N. Devilliers (1974), "Parameter Variation for the Solution of Two-Point Boundary Value Problems and Applications in the Calculus of Variations," JOTA, Vol. 13, pp. 164-178.

23. Hachtel, G. and M. Mack (1974), "A Pseudo-Dynamic Method for Solving Nonlinear Algebraic Equations," in *Stiff Differential Systems* (ed. R. A. Willoughby), New York: Plenum Press, pp. 135-150.

24. Hadamard, J. (1906), "Sur les Transformations Ponctuelles," Bull. Soc. Math. France, Vol. 34, pp. 71-84.

25. Jamshidi, M. (1972), "On the Embedding Solution of a Class of Optimal Control Problems," Automatica, Vol. 8, pp. 637-640.

26. Jamshidi, M., G. D'Ans and P. V. Kokotovic (1970), "Applications of a Parameter Imbedded Riccati Equation," IEEE Trans. Auto. Cont., Vol. AC-15, pp. 682-683.

27. Kalaba, R. E., E. Zagustin, W. Holbrow and R. Huss (1977), "A Modification of Davidenko's Method for Nonlinear Systems," Comp. and Math. with Appl., Vol. 3, pp. 315-319.

28. Keller, H. B. (1968), *Numerical Methods for Two-Point Boundary Value Problems*, Waltham, Mass.: Blaisdell.

29. Keller, H. B. (1970), "Nonlinear Bifurcation," J. Diff. Equations, Vol. 7, pp. 417-434.

30. Keller, H. B. (1971), "Shooting and Imbedding for Two-Point Boundary Value Problems," J. Math. Anal. Appl., Vol. 36, pp. 598-610.

31. Keller, H. B. (1975), "Approximation Methods for Nonlinear Problems with Application to Two-Point Boundary Value Problems," Math. of Comp., Vol. 29, pp. 464-474.

32. Keller, H. B. (1977), "Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems," in *Applications of Bifurcation Theory* (ed. P. H. Rabinowitz), New York: Academic Press.

33. Keller, H. B. and W. F. Langford (1972), "Iterations, Perturbations and Multiplicities for Nonlinear Bifurcation Problems," Arch. Rat. Mech. Anal., Vol. 48, pp. 83-108.

34. Keller, H. G. and A. Wolfe (1965), "On the Nonunique Equilibrium States and Buckling Mechanism of Spherical Shells," SIAM J. Appl. Math., Vol. 13, pp. 674-705.

35. Kellogg, R. B. T. Y. Li and J. A. Yorke (1976), "A Constructive Proof of the Brouwer Fixed Point Theorem and Computational Results," SIAM J. Numer. Anal., Vol. 13, pp. 473-483.

36. Kellogg, R. B., T. Y. Li and J. A. Yorke (1977), "A Method of Continuation for Calculating a Brouwer Fixed Point," in *Computing Fixed Points with Applications* (ed. S. Karamadian), New York: Academic Press, pp. 133-147.

37. Kizner, W. (1964), "A Numerical Method for Finding Solutions of Nonlinear Equations," SIAM J. Appl. Math., Vol. 12, pp. 424-428.

38. Klopfenstein, R. W. (1961), "Zeros of Nonlinear Functions," J. ACM, Vol. 8, pp. 366-373.

39. Kubicek, M. (1976), Algorithm 502, "Dependence of Solution of Nonlinear Systems on a Parameter," ACM-TOMS, Vol. 2, pp. 98-107.

40. Kubicek, M. and V. Hlavacek (1972a), "Solution of Nonlinear Boundary Value Problems--Va. A Novel Method: General Parameter Mapping (GPM)," Chem. Eng. Sci., Vol. 27, pp. 743-750.

41. Kubicek, M. and V. Hlavacek (1972b), "Solution of Nonlinear Boundary Value Problems--Vb. Predictor-Corrector GPM Method," Chem. Eng. Sci., Vol. 27, pp. 2095-2098.

42. Kubicek, M. and V. Hlavacek (1973), "General Parameter Mapping Technique--a Procedure for Solution of Nonlinear Boundary Value Problems Depending on an Actual Parameter," J. Inst. Math. Applic., Vol. 12, pp. 287-293.

43. Lahaye, E. (1934), "Une Methode de Resolution d'une Categorie d'Equations Transcendantes," C. R. Acad. Sci. Paris, Vol. 198, pp. 1840-1842.

44. Lahaye, E. (1935), "Sur la Representation des Racines Systemes d'Equations Transcendantes," Deuxieme Congres National des Sciences 1, pp. 141-146.

45. Lahaye, E. (1948), "Solution of Systems of Transcendental Equations," Acad. Roy. Belg. Bull. Cl. Sci., Vol. 5, pp. 805-822.

46. Lasota, A. and J. A. Yorke (1972), "Existence of Solutions of Two-Point Boundary Value Problems for Nonlinear Systems," J. Diff. Equations, Vol. 11, pp. 509-518.

47. Meyer, G. (1968), "On Solving Nonlinear Equations with a One-Parameter Operator Imbedding," SIAM J. Numer. Anal., Vol. 5, pp. 739-752.

48. Moore, R. H. (1964), "Newton's Method and Variations," in *Nonlinear Integral Equations* (ed. P. M. Anselone), U. of Wisconsin Press.

49. Orava, P. J. and P. A. J. Lautala (1976), "Back and Forth Shooting Method for Solving Two-Point Boundary Value Problems," JOTA, Vol. 18, pp. 485-498.

50. Orava, P. J. and P. A. J. Lautala (1977), "Interval Length Continuation Method for Solving Two-Point Boundary Value Problems," JOTA, Vol. 23, pp. 217-227.

51. Ortega, J. M. and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, New York: Academic Press.

52. Rall, L. B. (1968), "Davidenko's Method for the Solution of Nonlinear Operator Equations," U. of Wisconsin, Math. Research Center, MAC Tech. Summary Report 948.

53. Rheinboldt, W. C. (1969), "Local Mapping Relations and Global Implicit Function Theorems," Trans. Amer. Math. Soc., Vol. 138, pp. 183-198.

54. Rheinboldt, W. C. (1977), "Numerical Continuation Methods for Finite Element Applications," Proc. U.S.-German Symp. on Formulation and Computational Algorithms in Finite Element Analysis (ed. Bathe), MIT Press.

55. Roberts, S. and J. Shipman (1967), "Continuation in Shooting Methods for Two-Point Boundary Value Problems," J. Math. Anal. Appl., Vol. 18, pp. 45-58.

56. Roberts, S. and J. Shipman (1968), "Justification for the Continuation Method in Two-Point Boundary Value Problems," J. Math. Anal. Appl., Vol. 21, pp. 23-30.

57. Roberts, S. and J. Shipman (1972), *Two-Point Boundary Value Problems: Shooting Methods*, New York: American Elsevier.

58. Roberts, S., J. Shipman and C. V. Roth (1968), "Continuation in Quasi-linearization," JOTA, Vol. 2, pp. 164-178.

59. Simpson, R. B. (1975), "A Method for the Numerical Determination of Bifurcation States of Nonlinear Systems of Equations," SIAM J. Numer. Anal., Vol. 12, pp. 439-451.

60. Smale, S. (1976), "A Convergent Process of Price Adjustment and Global Newton Methods," J. Math. Econ., Vol. 3, pp. 1-14.

61. Thurston, G. A. (1969), "Continuation of Newton's Method Through Bifurcation Points," J. Appl. Mech., pp. 425-430.

62. Vaĭnberg, M. M. and V. A. Trenogin (1974), "Theory of Branching of Solutions of Nonlinear Equations," Leyden, Netherlands: Noordhoff International Publications.

63. Wasserstrom, E. (1971), "Solving Boundary-Value Problems by Imbedding," J. ACM, Vol. 18, pp. 594-602.

64. Wasserstrom, E. (1973), "Numerical Solutions by the Continuation Method," SIAM Review, Vol. 15, pp. 89-119.

65. Wasserstrom, E. and A. Kirszenblat (1973), "Analog Solutions of Nonlinear Boundary Value Problems by the Continuation Method," IEEE Trans. Computers, Vol. C-22, pp. 966-970.

CHAPTER 4
SPT APPROXIMATION OF OPTIMAL CONTROL LAWS

4.1 Introduction

For the purposes of on-board control of an aircraft we are not interested so much in obtaining optimal control trajectories as in obtaining optimal control laws. The essential difference between these two ways of specifying control is that the optimal control trajectory specifies the control to use as a function of time while the optimal control law specifies the control to use as a function of state. In principle, the control law is more valuable because it can provide feedback correction to perturbations in the optimal trajectory caused by uncertain environmental factors, e.g., wind gusts, or by errors in the mathematical model of the system, e.g., modeling the aircraft as a point mass rather than as a rigid body. Computationally, however, control trajectories are much easier to compute than optimal control laws because one need only calculate a function of one time variable rather than a function of several state variables. Indeed, even if we could calculate the optimal control law exactly off-line, the storage requirements would prohibit us from using this as an on-board control for any but the most modest sized problems. Suppose n is the number of states for the system, m is the number of controls, and N is the number of discrete values we divide any one of the n state variables into for storage. Then to store the control law requires storing an n-dimensional array with $n \times N^n$ elements. In our problem we have $n = 6$ and $m = 3$. Even if we were willing to accept the crude approximation of the control law resulting from using only $N = 10$ discrete values for each of the $n = 6$ state variables, we would still have to store a 6-dimensional array with 30×10^6 elements. Thus, the well-known "curse of dimensionality" forces us to use approximations for both the computation and the implementation of optimal control laws.

In this chapter we present some general methods for approximating optimal control laws by means of singular perturbation theory (SPT) techniques. In the next chapter we apply these general methods to the minimum time control of the six-state aircraft model described in Chapter 2. A brief summary of the contents of the present chapter follows.

Section 4.2 describes the basic calculation of the SPT approximation to the optimal control law for a general dynamic system. To clarify this calculation we discuss in detail the case of two time scales. The general case of several time scales is discussed more briefly--we present the results and leave the details of the derivation to an appendix. Section 4.3 discusses the computational difficulties of the SPT calculation described in Section 4.2 and it also discusses alternative strategies for overcoming these difficulties in order to obtain efficient real-time algorithms. In particular, we discuss in detail the use of individual time scales for each individual state variable, linearization of the SPT calculation outside of regions of rapid variation and the use of suboptimal solutions for the slow time scales. Finally, Section 4.4 discusses the problem of the validity of the SPT approximation in various regions of state space. For example, by properly scaling the aircraft equations we obtain a natural singular perturbation parameter that is inversely proportional to total range in the interception problem. In this case, the SPT approximation is better the farther the state is from the terminal target states. Near the terminal target the SPT approximation breaks down dramatically and implementation of the SPT control law leads to a destabilizing feedback law. Thus, in the vicinity of the target state one must switch from SPT to some other method of approximating the control law.

4.2 Calculation of the SPT Approximation

4.2.1 Control Trajectories, Control Laws and the SPT Approximation

In Section 4.2 we will discuss the SPT approximation of optimal control laws of the following generic, autonomous control problem. Let x represent an n-dimensional state vector trajectory and let u represent an m-dimensional control vector trajectory. The dynamic equations for the system are:

$$\frac{dx}{dt} = f(x,u) \tag{2.1}$$

Suppose that u is a given control input trajectory and let x be the corresponding state trajectory obtained from solving the differential equation (2.1) over the time interval $[t_0,\infty)$ with the initial condition

$$x(t_0) = \xi \tag{2.2}$$

The cost functional J(u) corresponding to the given input u is defined as the integral cost criterion

$$J(u) \triangleq \int_{t_0}^{t_f} L(x,u)\, dt \tag{2.3}$$

where t_f represents the final time at which we require the state to lie in the terminal target set. For simplicity we assume that the terminal target set consists of the single state ω so that the terminal condition for the problem is

$$x(t_f) = \omega \tag{2.4}$$

In this generic control problem we will allow the terminal time t_f to be either free or fixed and we will specify which is the case unless the results are the same for either case.

The optimal control problem is to find a control input u* such that (2.3) is minimized by $u = u^*$ subject to the condition that (2.1), (2.2) and (2.4) also hold. For a given fixed terminal time t_f and a given fixed terminal condition (2.4), the optimal input u* is a function of the initial condition ξ, the initial time t_0 and the current time t where $t_0 \leq t \leq t_f$. That is, we have

$$u^* = u(t,t_0,\xi) \tag{2.5}$$

By fixing the initial time t_0 and the initial state ξ, we can consider the optimal control in (2.5) as a function of the current time t only, i.e., as a <u>control trajectory</u>. On the other hand, if we let the initial time t_0 and the initial state ξ vary and if we set $t = t_0$, then we can consider the optimal control in (2.5) as a function of time and state only, i.e., as a <u>control law</u>. Here we are interested in obtaining an SPT approximation of the control law and we will do this as follows. Using the SPT methods of trajectory optimization, we will approximate the optimal control function in (2.5) for arbitrary initial times and initial states. We will then set $t = t_0$ in (2.5) to obtain the optimal control u* to use at time t_0 with the corresponding state ξ. Since t_0 and ξ were arbitrary, this procedure will give us an approximate control

law to use at any time for any state for which the SPT approximation is valid. In Section 4.4 we will see that the SPT approximation is not valid for certain values of ξ.

Suppose that the control problem in (2.1) - (2.4) has only two time scales, fast and slow, and suppose that $x = (x^1, x^2)$ where x^1 represents the fast components of x and x^2 represents the slow components of x. Then the singular perturbation approximation to (2.1) is

$$\frac{dx^1}{dt} = f^1(x,u) \tag{2.6}$$

$$\varepsilon\frac{dx^2}{dt} = f^2(x,u) \tag{2.7}$$

where ε is a parameter varying from 0 to 1 and f^i is the component of f corresponding to the x^i component of x, $i = 1,2$. For each ε the optimal control problem with (2.6),(2.7), (2.3),(2.4) results in a TPBVP similar to the unperturbed ($\varepsilon = 1$) problem. In terms of the Hamiltonian H defined

$$H(x,\lambda,u) = L(x,u) + \lambda^T f(x,u) \tag{2.8}$$

where $\lambda = (\lambda^1, \lambda^2)$, we have the perturbed equations

$$\frac{d\lambda^1}{dt} = -\frac{\partial H}{\partial x^1} \tag{2.9}$$

$$\frac{d\lambda^2}{dt} = -\frac{\partial H}{\partial x^2} \tag{2.10}$$

which correspond to (2.6) and (2.7). The SPT approximation of the control trajectory results from approximating the TPBVP given by (2.6), (2.7), (2.9) and (2.10). Typically, this approximation has three parts: (1) an initial boundary layer at the initial time t_0, (2) a reduced order, or outer, solution for t such that $t_0 < t < t_f$, and (3) a terminal boundary layer at the terminal time t_f. These different parts are determined as follows. The initial boundary layer is given by solving

(2.11) $$\frac{d\hat{x}^1}{d\tau} = 0$$

(2.12) $$\frac{d\hat{x}^2}{d\tau} = f^2(\hat{x},\hat{u})$$

(2.13) $$\frac{d\hat{\lambda}^1}{d\tau} = 0$$

(2.14) $$\frac{d\hat{\lambda}^2}{d\tau} = -\frac{\partial H}{\partial x^2}(\hat{x},\hat{\lambda},\hat{u})$$

where τ is the fast time scale given by $\tau = \frac{t-t_0}{\varepsilon}$. The initial conditions for this TPBVP are the same as the initial conditions (2.2) for the unperturbed ($\varepsilon = 1$) problem. That is,

(2.15) $$\hat{x}^1(0) = \xi^1$$

(2.16) $$\hat{x}^2(0) = \xi^2$$

The terminal conditions are not those for the original problem, however. Instead we have the asymptotic terminal conditions

(2.17) $$\lim_{\tau\to\infty} \hat{x}^1(\tau) = \bar{x}^1(0)$$

(2.18) $$\lim_{\tau\to\infty} \hat{x}^2(\tau) = \bar{x}^1(0)$$

(2.19) $$\lim_{\tau\to\infty} \hat{\lambda}^1(\tau) = \bar{\lambda}^1(0)$$

(2.20) $$\lim_{\tau\to\infty} \hat{\lambda}^2(\tau) = \bar{\lambda}^2(0)$$

where $\bar{x}^1$, $\bar{x}^2$, $\bar{\lambda}^1$, $\bar{\lambda}^2$ are the states and adjoints for the reduced order solution. That is, $\bar{x}^1$, $\bar{x}^2$, $\bar{\lambda}^1$, $\bar{\lambda}^2$ satisfy the TPBVP

(2.21) $$\frac{d\bar{x}^1}{dt} = f^1(\bar{x},\bar{u})$$

(2.22) $$0 = f^2(\bar{x},\bar{u})$$

$$(2.23) \qquad \frac{d\bar{\lambda}^1}{dt} = -\frac{\partial H}{\partial x^1}(\bar{x},\bar{\lambda},\bar{u})$$

$$(2.24) \qquad 0 = -\frac{\partial H}{\partial x^2}(\bar{x},\bar{\lambda},\bar{u})$$

The initial condition is

$$(2.25) \qquad \bar{x}^1(0) = \xi^1$$

and the final condition is given by

$$(2.26) \qquad \bar{x}^1(t_f) = \omega^1$$

Note that the reduced order solution treats $\bar{x}^2$ as a pseudocontrol which satisfies the algebraic equation (2.22) rather than the original differential equation. Finally, the terminal boundary layer is given by solving

$$(2.27) \qquad \frac{d\tilde{x}^1}{d\sigma} = 0$$

$$(2.28) \qquad \frac{d\tilde{x}^2}{d\sigma} = f^2(\tilde{x},\tilde{u})$$

$$(2.29) \qquad \frac{d\tilde{\lambda}^1}{d\sigma} = 0$$

$$(2.30) \qquad \frac{d\tilde{\lambda}^2}{d\sigma} = -\frac{\partial H}{\partial x^2}(\tilde{x},\tilde{\lambda},\tilde{u})$$

where σ is the fast time scale $\sigma = \frac{t-t_f}{\varepsilon}$. The initial conditions for this TPBVP are the same as the <u>final</u> conditions (2.4) of the original problem. That is,

$$(2.31) \qquad \tilde{x}^1(0) = \omega^1$$

$$(2.32) \qquad \tilde{x}^2(0) = \omega^2$$

The terminal conditions are given asymptotically in terms of the reduced order

solution as

$$\lim_{\sigma \to -\infty} \tilde{x}^1(\sigma) = \bar{x}^1(t_f) \tag{2.33}$$

$$\lim_{\sigma \to -\infty} \tilde{x}^2(\sigma) = \bar{x}^2(t_f) \tag{2.34}$$

$$\lim_{\sigma \to -\infty} \tilde{\lambda}^1(\sigma) = \bar{\lambda}^1(t_f) \tag{2.35}$$

$$\lim_{\sigma \to -\infty} \tilde{\lambda}^2(\sigma) = \bar{\lambda}^2(t_f) \tag{2.36}$$

After solving (2.11) - (2.36) we obtain three control trajectories: the initial boundary layer trajectory $\hat{u}(\tau)$, the reduced order trajectory $\bar{u}(t)$ and the terminal boundary layer $\tilde{u}(\sigma)$. The SPT approximation which is uniformly valid over the whole interval $t_0 \leq t \leq t_f$ is given by

$$u^*(t,t_0,\xi) \simeq [\hat{u}(\tau) - \bar{u}(0)] + [\tilde{u}(\sigma) - u(t_f)] \tag{2.37}$$

where $\tau = \frac{t-t_0}{\varepsilon}$ and $\sigma = \frac{t-t_f}{\varepsilon}$. Note that we have suppressed the dependence of $\hat{u}$, $\bar{u}$ and $\tilde{u}$ on t_0 and ξ for clarity. Here we have only sketched the SPT method of approximating solution trajectories of the TPBVP one obtains for an optimal control problem. In Subsection 4.2.2 we will present this approximation in more detail for the problem of obtaining control law approximations. The reader should refer to Kelley (1973) for more details about obtaining optimal trajectory approximations by means of SPT.

Because we only wish to obtain the approximation of the time function $t \to u^*(t,t_0,\xi)$ at the initial time $t = t_0$, we only need to obtain the initial boundary layer terms in the SPT approximation. However, to obtain this term we need to calculate the reduced order approximation first. *Nevertheless, we do not need to consider the terminal boundary layer at all to obtain the lowest order approximation.* To obtain the higher order terms of the initial boundary layer we would have to calculate higher order terms of both the reduced order approximation and the terminal boundary layer approximation. The following discussion will help clarify this situation.

Equation (2.37) represents the first term in an asymptotic approximation

for the optimal control trajectory u* as a power series in ε. The full asymptotic series has the form

$$(2.38) \qquad u^* \simeq \sum_{k=0}^{\infty} \{[\hat{u}_k(\tau) - \bar{u}_k(t_0)] + \bar{u}_k(t) + [\tilde{u}_k(\sigma) - \bar{u}_k(t_f)]\}\varepsilon^k$$

where $\hat{u}_k$ denotes the $k^{\underline{th}}$ order term of the initial boundary layer approximation, $\bar{u}_k$ denotes the $k^{\underline{th}}$ order term of the reduced order solution and $\tilde{u}_k$ denotes the $k^{\underline{th}}$ order term of the terminal boundary layer approximation. Note that we have suppressed the ξ dependence of the functions $\hat{u}_k$, $\bar{u}_k$, $\tilde{u}_k$ in order to maintain notational clarity.

The boundary layer terms $\hat{u}_k$ and $\tilde{u}_k$ in (2.38) must have the following asymptotic properties:

$$(2.39) \qquad \lim_{\tau \to \infty} \hat{u}_k(\tau) = \bar{u}_k(t_0)$$

$$(2.40) \qquad \lim_{\sigma \to -\infty} \tilde{u}_k(\sigma) = \bar{u}_k(t_f)$$

Moreover, for reasonably well-behaved problems the convergence in (2.39) and (2.40) will be exponential. That is, for some positive constants a_k, b_k we will have

$$(2.41) \qquad |\hat{u}_k(\tau) - \bar{u}_k(t_0)| \leq e^{-a_k \tau}$$

$$(2.42) \qquad |\tilde{u}_k(\sigma) - \bar{u}_k(t_f)| \leq e^{-b_k |\sigma|}$$

for sufficiently large τ and $|\sigma|$. Using the information (2.42) and evaluating (2.38) at $t = t_0$, we obtain

$$(2.43) \qquad u^* \simeq \sum_{k=0}^{\infty} \{\hat{u}_k(0)\varepsilon^k + O\left(e^{-b_k(t_f - t_0)/\varepsilon}\right)\varepsilon^k\}$$

where, as usual, the order notation $O(\phi(\varepsilon))$ denotes a function which decreases as fast as $\phi(\varepsilon)$ as $\varepsilon \to 0$. The SPT approximation in (2.38) is only an asymptotic approximation as $\varepsilon \to 0$. Moreover, in (2.43) the exponential terms

$O\left(e^{-b_k(t_f-t_0)/\varepsilon}\right)\varepsilon^k$ decrease exponentially faster than the terms $\hat{u}_k(0)\varepsilon^k$. Thus, the exponential terms are negligible in the asymptotic expansion, and we have

$$u^* \simeq \sum_{k=0}^{\infty} \hat{u}_k(0)\varepsilon^k \tag{2.44}$$

From (2.44) we see that in order to obtain the $k^{\underline{th}}$ order approximation for u^* we need only find the $k^{\underline{th}}$ order initial boundary layer approximation

$$\sum_{j=0}^{k} u_j(0)\varepsilon^j \tag{2.45}$$

However, note that to obtain the term $\hat{u}_0$ in (2.45) we need to calculate the reduced order term $\bar{u}_0$ first. To obtain the next term $\hat{u}_1$ we must calculate $\tilde{u}_0$ and $\bar{u}_1$ as well. In general, to obtain the $k^{\underline{th}}$ order initial boundary layer approximation we have to calculate the reduced order approximation up to the $k^{\underline{th}}$ order and the terminal boundary layer approximation up to the $(k-1)^{\underline{th}}$ order. The important point to note is that to find the $0^{\underline{th}}$ order SPT approximation, the one which we will be using to approximate the aircraft control law in the succeeding chapters, requires only the calculation of the initial boundary layer and the reduced order approximation of the $0^{\underline{th}}$ order--the terminal boundary layer approximation does not enter the calculation at all.

The expression (2.43) indicates when the SPT approximation breaks down; namely, when the initial time t_0 and the final time t_f are very close together so that $(t_0-t_f)/\varepsilon$ is small and the exponential terminal boundary layer terms in (2.43) cannot be neglected. In this case, the SPT approximation of u^* in terms of initial boundary layer, reduced order and terminal boundary layer approximations is invalid and the asymptotic approximation in (2.38) is incorrect. In particular, one cannot improve (2.44) by including the terminal boundary layer terms in (2.38). We discuss this situation further and illustrate it with simple examples in Section 4.4. In the next subsection we discuss the details of the calculation of the $k = 0$ term of (2.44) for the two time scale case.

4.2.2 Calculation of the SPT Approximation: Two Time Case

Although the aircraft example considered in this project is treated as a

system with more than two time scales, the notational difficulties of presenting the multi-time scale case obscure the basic simplicity of the SPT calculation of the approximate optimal control. Therefore, to illustrate the basic method we will discuss the case of a two time scale dynamical system in some detail in this subsection. In the next subsection, 4.2.3, we will present and discuss the results for the multi-time case, but we leave the details of the calculation to Appendix 4.1 at the end of Chapter 4.

Consider the dynamic system

$$\frac{dx}{dt} = f(x,y,u) \tag{2.46}$$

$$\varepsilon\frac{dy}{dt} = g(x,y,u) \tag{2.47}$$

where $x \in X$, $y \in Y$ and $u \in U$. We assume that X, Y, U are vector spaces which are not necessarily one-dimensional. The cost criterion for the control problem is

$$J(u) \triangleq \int_0^{t_f} L(x,y,u)\, dt \tag{2.48}$$

where t_f is a terminal time which may be either fixed or free. The terminal conditions for the problem are

$$x(t_f) = x_f \tag{2.49}$$

$$y(t_f) = y_f \tag{2.50}$$

and the initial conditions for the problem are

$$x(0) = \xi \tag{2.51}$$

$$y(0) = \eta \tag{2.52}$$

To find the first term of the asymptotic series (2.44) we need to obtain the initial boundary layer approximation, but as we indicated in the previous section, we must first calculate the reduced order approximation. The reduced order approximation to the system (2.46) - (2.52) is obtained by setting the perturbation parameter $\varepsilon = 0$ in (2.47) and omitting the initial condition (2.50)

and the terminal condition (2.52). If the quantities $\bar{x}$, $\bar{y}$, $\bar{u}$ denote the reduced order approximation, then we obtain these quantities by solving the problem with the dynamic system

$$(2.53) \qquad \frac{d\bar{x}}{dt} = f(\bar{x},\bar{y},\bar{u})$$

where $\bar{y}$ and $\bar{u}$ are both considered as controls for the state $\bar{x}$. In addition, we require that the following equality constraint be satisfied:

$$(2.54) \qquad 0 = g(\bar{x},\bar{y},\bar{u})$$

The cost criterion for the reduced order problem is the same as for the original problem (2.48), namely

$$(2.55) \qquad \bar{J}(\bar{y},\bar{u}) = \int_0^{t_f} L(\bar{x},\bar{y},\bar{u})\, dt$$

In the reduced order problem only the initial and terminal conditions for x are retained. Thus, we have the initial condition

$$(2.56) \qquad \bar{x}(0) = \xi$$

and the terminal condition

$$(2.57) \qquad \bar{x}(t_f) = x_f$$

To solve the reduced order problem we define the reduced order Hamiltonian

$$(2.58) \qquad \bar{H}(\bar{x},\bar{y},\bar{u},\bar{\lambda}_x) = L(\bar{x},\bar{y},\bar{u}) + \bar{\lambda}_x f(\bar{x},\bar{y},\bar{u})$$

in which $\bar{\lambda}_x$ denotes a reduced order adjoint variable for the state $\bar{x}$. The minimum principle implies that the optimal controls $\bar{y}^*$ and $\bar{u}^*$ are chosen so that on the optimal trajectory we have

$$(2.59) \qquad \bar{H}(\bar{x}(t),\bar{y}^*(t),\bar{u}^*(t),\bar{\lambda}_x(t)) = \min_{\bar{y},\bar{u}} H(\bar{x}(t),\bar{y},\bar{u},\bar{\lambda}_x(t))$$

Provided that we can solve the minimization problem in (2.27) for the controls

$\bar{y}^*$ and $\bar{u}^*$ in terms of $\bar{x}$ and $\bar{\lambda}_x$, we have the following two-point boundary value problem (TPBVP) for the $\bar{x}$ and $\bar{\lambda}$ trajectories:

$$(2.60) \qquad \frac{d\bar{x}}{dt} = f(\bar{x},\bar{y}^*(\bar{x},\bar{\lambda}_x),\bar{u}^*(\bar{x},\bar{\lambda}_x))$$

$$(2.61) \qquad \frac{d\bar{\lambda}_x}{dt} = -\frac{\partial H}{\partial \bar{x}}(\bar{x},\bar{y}^*(\bar{x},\bar{\lambda}_x),\bar{u}^*(\bar{x},\bar{\lambda}_x),\bar{\lambda}_x)$$

The boundary conditions for this TPBVP are the initial and terminal conditions on $\bar{x}$ given in (2.56) and (2.57). If the terminal time t_f is free, then there is an additional condition on the Hamiltonian, namely,

$$(2.62) \qquad \bar{H}(\bar{x}(t),\bar{y}(t),\bar{u}(t),\bar{\lambda}_x(t)) = 0$$

for all times t such that $0 \le t \le t_f$.

In order to obtain the initial boundary layer we need to have the optimal reduced order adjoint $\bar{\lambda}_x^*$ and the optimal reduced order pseudocontrol $\bar{y}^*$ at the initial time $t = 0$. Assuming that the terminal state (2.57) is fixed and that the initial condition (2.56) is variable, then the adjoint $\bar{\lambda}_x^*$ and the pseudocontrol $\bar{y}^*$ at the initial time $t = 0$ depend on the initial state ξ and the final time t_f as follows:

$$(2.63) \qquad \bar{\lambda}_x^* = \bar{\lambda}_x^*(\xi,t_f)$$

$$(2.64) \qquad \bar{y}^* = \bar{y}^*(\xi,t_f)$$

If the final time is free, then the adjoint and control at the initial time $t = 0$ depend only on the initial state ξ. Having obtained (2.63) and (2.64), we can now formulate the initial boundary layer problem.

The initial boundary layer problem is obtained from (2.46) and (2.47) by transforming these equations to the fast time scale $\tau = t/\varepsilon$. Thus, we have

$$(2.65) \qquad \frac{d\hat{x}}{d\tau} = \varepsilon f(\hat{x},\hat{y},\hat{u})$$

$$(2.66) \qquad \frac{d\hat{y}}{d\tau} = g(\hat{x},\hat{y},\hat{u})$$

where $\hat{x}$, $\hat{y}$, $\hat{u}$ denote the initial boundary layer approximation to the quantities x, y and u. Note that in this approximation both $\hat{x}$ and $\hat{y}$ are states and $\hat{u}$ is the only control. Since we are only interested in the $0^{\underline{th}}$ order approximation, we set $\varepsilon = 0$ in (2.65) to obtain

$$\frac{d\hat{x}}{d\tau} = 0 \tag{2.67}$$

$$\frac{d\hat{y}}{d\tau} = g(\hat{x},\hat{y},\hat{u}) \tag{2.68}$$

Thus, the state $\hat{x}$ is constant given by the initial condition (2.51),

$$\hat{x}(\tau) = \xi \tag{2.69}$$

for all $\tau \geq 0$. Likewise, the adjoint $\hat{\lambda}_x$ corresponding to $\hat{x}$ is constant and is given by

$$\hat{\lambda}_x(\tau) = \bar{\lambda}_x(\xi,t_f) \tag{2.70}$$

for all $\tau \geq 0$. Using (2.69) and (2.70), we may rewrite the initial boundary layer problem as an infinite horizon control problem for $\hat{y}$ and $\hat{u}$ alone. The dynamic system for this control problem is

$$\frac{d\hat{y}}{d\tau} = g(\xi,\hat{y},\hat{u}) \tag{2.71}$$

with the initial condition

$$\hat{y}(0) = \eta \tag{2.72}$$

which is derived from (2.52). The terminal condition for the problem is given as an asymptotic limit, namely

$$\lim_{\tau\to\infty} \hat{y}(\tau) = \bar{y}(\xi,t_f) \tag{2.73}$$

This asymptotic terminal condition is the requirement that the initial boundary layer match up with the reduced order approximation. The infinite horizon cost criterion for the initial boundary layer problem is given in terms of the cost

function L and the reduced order adjoint $\bar{\lambda}_x$ as

$$(2.74) \qquad \hat{J}(\hat{u}) = \int_0^\infty [L(\xi,\hat{y},\hat{u}) + \bar{\bar{\lambda}}_x(\xi,t_f)f(\xi,\hat{y},\hat{u})]\, d\tau$$

The optimal control trajectory $\hat{u}^*$ for the initial boundary layer problem depends directly on the initial condition (2.72) and the fast time τ, and indirectly on the initial condition ξ and the final time t_f for the reduced order problem. Thus,

$$(2.75) \qquad \hat{u}^* = \hat{u}^*(\tau,\eta,\xi,t_f)$$

As we saw in (2.44), we only need (2.75) at $\tau = 0$ for the SPT approximate optimal control law. Thus, the approximate optimal feedback control law u^* is

$$(2.76) \qquad u^* \simeq \hat{u}^*(0,\eta,\xi,t_f)$$

The approximation (2.76) expresses the control u^* as a feedback function of the current state (ξ,η) and the time-to-go t_f.

In terms of the preceding discussion the calculation of the SPT approximate optimal control law can be summarized as follows (also see Figure 4.2.1): for a given state (ξ,η) and time-to-to t_f

(i) calculate the optimal control adjoint and the pseudocontrol law for the reduced order problem in (2.53) through (2.57). Note that the adjoint and the pseudocontrol for the reduced order problem will depend on the current x state ξ and the time-to-go t_f.

(ii) Using the adjoint and pseudocontrol from the reduced order approximation to define the cost and the asymptotic terminal condition for the initial boundary layer problem, calculate the optimal control law for the initial boundary layer problem (2.71) through (2.74). Note that this solution will depend on the current y state η directly, and indirectly on the current x state ξ and time-to-go t_f. The current x state and the time-to-go enter the initial boundary layer problem only in the asymptotic terminal condition (2.73) and the cost functional (2.74).

Before turning to the general multi-time scale case, let us note that without further assumptions the SPT approximation is no easier to solve than the

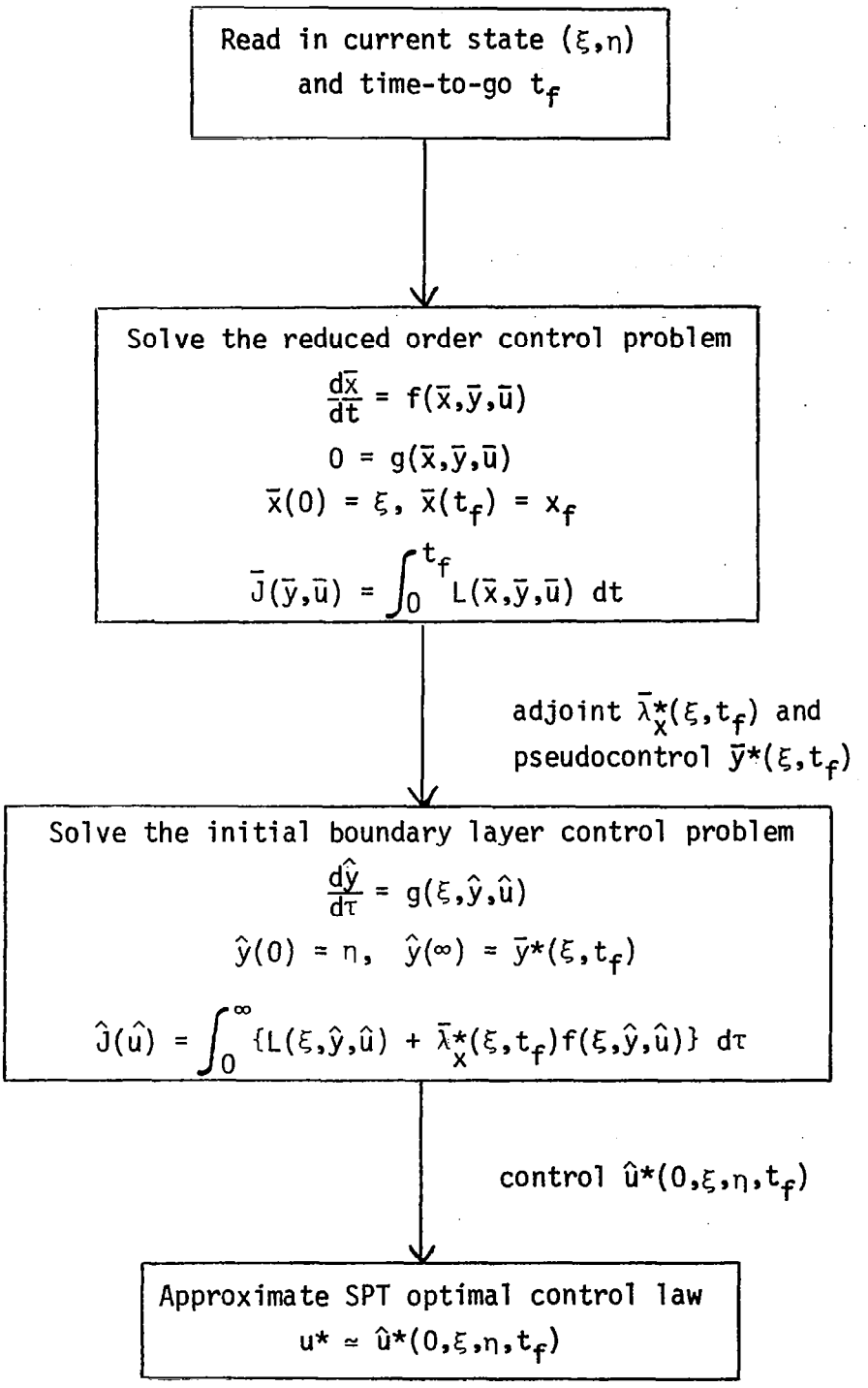


Figure 4.2.1.-General SPT Algorithm: Two Time Scales.

original control problem in (2.46) through (2.52). Although the reduced order problem in $\bar{x}$ is a smaller dimensional problem than the original problem in x and y, the initial boundary layer problem in $\hat{y}$ could be as difficult to solve as the original problem due to the general cost functional (2.74) and the general terminal condition (2.73) which depend on the current x state ξ. We will discuss this point in more detail in Section 4.3.

4.2.3 Calculation of the SPT Approximation: Multi-Time Case

The calculation of the SPT approximation to the optimal control law in the multi-time case is conceptually the same as in the two-time case. Therefore, we will present only the basic algorithm for the calculation in this section and leave the details of the formulation and calculation to Appendix 4.1 at the end of Chapter 4. Consider the generic, autonomous control system introduced in Subsection 4.2.1 in equations (2.1) through (2.4). Let us suppose that the state vector x has been decomposed into lower dimensional components x^i of dimension n_i for $i = 0,1,\ldots,r$. If x has dimension n, then $0 \le r \le n-1$ and for each i we must have $1 \le n_i \le n$. Each component corresponds to a separate time scale with x_0 representing the slowest time scale and x_r representing the fastest time scale. The time scales are arranged in order so that for each i the time scale for x_{i+1} is faster than the time scale for x_i. We now present the algorithm for calculating the SPT approximation to the optimal control in this case of r time scales. The reader may refer to Appendix 4.1 for details of the singular perturbation theory formulation and derivation.

To facilitate our presentation we introduce the following notation. A superscript i will always refer to a vector component corresponding to the x^i components of the x vector. Thus, we write f^i for the components of the vector function f in (2.1). On the other hand, the subscript i will refer to quantities corresponding to the $i^{\underline{th}}$ boundary layer problem. Thus, we define the pseudo-control u_i as

$$u_i = (x^{i+1},\ldots,x^r,u) \tag{2.77}$$

for $i = 0,1,\ldots,r-1$ and

$$u_r = u. \tag{2.78}$$

Similarly, we define ξ_i corresponding to the first i+1 initial conditions,

$$(2.79) \qquad \xi_i = (\xi^0,\ldots,\xi^i)$$

for $i = 0,1,\ldots,r$.
The vector functions g_i of (x,u) are defined so that

$$(2.80) \qquad g_i(x,u) = (f^{i+1}(x,u),\ldots,f^r(x,u))$$

for $i = 0,1,\ldots,r-1$ and so that

$$(2.81) \qquad g_r(x,u) = 0$$

The function g_{r+1}, which is trivially 0, is used only for notational convenience.

Algorithm for SPT Approximate Optimal Control Law*

The control to use with current state ξ and time-to-go t_f is calculated as follows:

(i) Solve the reduced order problem:

$$(2.82) \qquad \frac{dx^0}{dt} = f^0(x^0,u_0)$$

$$(2.83) \qquad 0 = g_0(x^0,u_0)$$

$$(2.84) \qquad x^0(0) = \xi^0$$

$$(2.85) \qquad x^0(t_f) = \omega^0$$

where ω is the terminal condition (2.4) for the original problem,

$$(2.86) \qquad J_0(u_0) = \int_0^{t_f} L(x^0,u_0)\,dt$$

where L is the cost criterion (2.3) in the original problem.
(ii) Let $\bar{x}^1$ denote the first component of the pseudocontrol u_0, and let $\bar{\lambda}_0$

*Also see Figure 4.2.2.

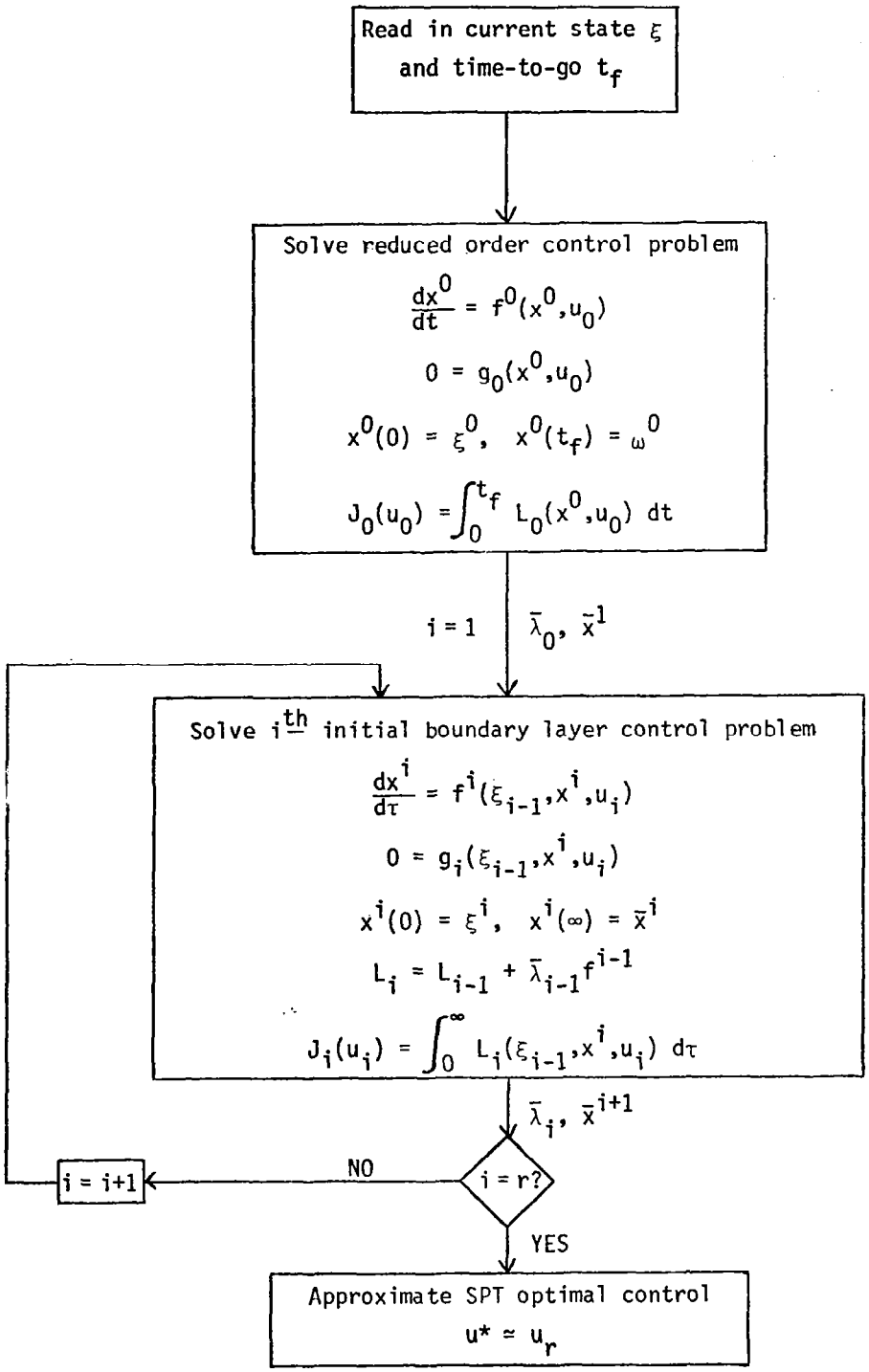


Figure 4.2.2.-General SPT Algorithm: Multi-Time Scales.

denote the adjoint variable of x^0 in the reduced order problem. Note that $\bar{x}^1$ and $\bar{\lambda}_0$ have the following functional dependence:

$$\bar{x}^1 = \bar{x}^1(\xi_0, t_f) \tag{2.87}$$

$$\bar{\lambda}_0 = \bar{\lambda}_0(\xi_0, t_f) \tag{2.88}$$

Let L_0 denote the cost criterion L in (2.86). Set $i = 1$ and go to the next step.

(iii) Solve the $i^{\underline{th}}$ initial boundary layer problem:

$$\frac{dx^i}{d\tau} = f^i(\xi_{i-1}, x^i, u_i) \tag{2.89}$$

$$0 = g_i(\xi_{i-1}, x^i, u_i) \tag{2.90}$$

$$x^i(0) = \xi^i \tag{2.91}$$

$$x^i(\infty) = \bar{x}^i(\xi_{i-1}, t_f) \tag{2.92}$$

$$J_i(u_i) = \int_0^\infty L_i(\xi_{i-1}, x^i, u_i)\, d\tau \tag{2.93}$$

where the cost criterion L_i for the $i^{\underline{th}}$ initial boundary layer problem is defined recursively in terms of L_{i-1} as

$$L_i(\xi_{i-1}, x^i, u_i) = L_{i-1}(\xi_{i-2}, \xi^{i-1}, x^i, u_i) + \bar{\lambda}_{i-1}(\xi_{i-1}, t_f) f^{i-1}(\xi_{i-1}, x^i, u_i) \tag{2.94}$$

(iv) If $i < r$, denote the first component of the pseudocontrol u_i calculated in step (iii) by $\bar{x}^{i+1}$ and let $\bar{\lambda}_i$ denote the adjoint corresponding to x^i in the $i^{\underline{th}}$ boundary layer problem. Update i by 1 and return to the beginning of step (iii).

Stop if $i = r$; $u = u_r$ gives the SPT approximate optimal feedback control for the current state ξ with time-to-go t_f.

The procedure for calculating the $0^{\underline{th}}$ order SPT approximation of the optimal control law consists of solving a hierarchy of lower dimensional optimal control

problems. At the $i = 0$ level of the hierarchy, which corresponds to the slowest component of the state variable, we must solve the reduced order control problem. This is an n_0 dimensional problem with finite terminal time, which may be free or fixed. At the subsequent levels for $i > 0$, which correspond to faster components of the state variable, we must solve the $i^{\underline{th}}$ initial boundary layer control problem. This is an n_i dimensional problem with infinite terminal time, i.e., it is an infinite horizon control problem.

At each level $i > 0$ of the hierarchy the cost criterion and the asymptotic terminal condition for the control problem is defined recursively from the previous level by the relation (2.94). Thus, the cost control problem at the $i^{\underline{th}}$ level of the hierarchy depends on the components ξ^j, for $0 \le j \le i$ of the current state ξ.

Once the final level of the hierarchy is reached, the SPT approximation to the optimal control for the current state ξ and the time-to-go t_f is the optimal control $u = u_r$ one obtains from this level.

4.3 Computational Aspects of the SPT Approximation

4.3.1 Computational Difficulties of SPT Approximation

In Subsection 4.2.3 we decomposed the original optimal control problem (2.1) - (2.4) of state dimension n into r+1 subproblems of dimension n_i for the $i^{\underline{th}}$ subproblem, $i = 0,1,\ldots,r$. Since we will have $n_i < n$ for each i, the $i^{\underline{th}}$ subproblem should be computationally easier to solve than the original n dimensional problem. However, the r+1 subproblems are coupled in such a way that the $i^{\underline{th}}$ problem requires results from the $(i-1)^{\underline{th}}$ problem to define the cost and terminal condition in the $i^{\underline{th}}$ level problem. If we define the integers N_i as the sums

$$N_i = \sum_{j=0}^{i} n_j \tag{3.1}$$

then the control subproblem at level i depends on N_i+1 parameters (or N_i parameters if the terminal time for the original problem is free). These parameters consist of the current state components ξ^j for $0 \le j \le i$ and the time-to-go t_f. At the $r^{\underline{th}}$ level, the final level, the control subproblem depends on

$N_r + 1 = n + 1$ parameters--namely, the entire current state ξ and the time-to-go t_f. Thus, the final level of the control subproblem depends on as many parameters as the original problem and may be as difficult to solve. Moreover, it is essential to compute this final subproblem in order to obtain a realistic feedback control for the original control problem. For if we stop the computation process at a level $i < r$, then we implicitly will be assuming that we can directly control the states x^j for $j > i$. In practice this means that the u-controls resulting from the i^{th} level will not provide feedback correction for these higher level states x^j for $j > i$.

Thus, the SPT approximation presents us with a dilemma. To obtain a feedback control law that will correct deviations from optimal of all state components we need to compute the final level of all the subproblems up to and including the final level. But to obtain a computational advantage over solving the original problem exactly (without SPT) by using the SPT approximation we need to stop our computation at some level before the final level. In the following three subsections we discuss three different strategies for making the SPT algorithm computationally efficient.

4.3.2 Complete Time Scale Separation

One possibility for making the SPT algorithm more efficient is to subdivide the original problem into a large number of small dimensional subproblems and to develop fast algorithms for solving these small dimensional optimal control problems with general cost criteria and general terminal conditions. The extreme case of this procedure is to choose $r = n-1$ so that $n_i = 1$ for each $i = 0,1,\ldots,r$. This is the approach Calise (1977, 1978) has taken in constructing feedback controls for aircraft by means of singular perturbation theory methods. Making each subproblem one-dimensional allows one to substitute a sequence of static optimization problems for the original dynamic optimization problem. The drawback of this procedure is that the assumption that each one-dimensional component of the state operates on its own time scale will be unrealistic when some components of the state are closely coupled together.

The transformation of the dynamic control problem to a static optimization problem is based on the following simple observation. Suppose that we have a dynamical system with a one-dimensional state variable x and dynamic equation

given by

$$\frac{dx}{d\tau} = f(x,u) \tag{3.2}$$

and cost criterion given by

$$J(u) = \int_0^\infty L(x,u)\, d\tau \tag{3.3}$$

The minimum principle asserts that the optimal adjoint variable λ_x^* for a given state x is chosen so that we have

$$c = \min_u \{L(x,u) + \lambda_x^* f(x,u)\} \tag{3.4}$$

where c is a constant not depending on x. From (3.4) we see that if u is a control such that $f(x,u) > 0$ then

$$\frac{c - L(x,u)}{f(x,u)} \leq \lambda_x^* \tag{3.5}$$

and if u is a control such that $f(x,u) < 0$ then

$$\frac{c - L(x,u)}{f(x,u)} \geq \lambda_x^* \tag{3.6}$$

Defining m(x) and M(x) so that for each x we have

$$m(x) = \max_u \left\{ \frac{c - L(x,u)}{f(x,u)} : \quad f(x,u) > 0 \right\} \tag{3.7}$$

$$M(x) = \min_u \left\{ \frac{c - L(x,u)}{f(x,u)} : \quad f(x,u) < 0 \right\} \tag{3.8}$$

we obtain the following upper and lower bounds on the adjoint λ_x^*:

$$m(x) \leq \lambda_x^* \leq M(x) \tag{3.9}$$

If the optimal control u* for the given state x is such that $f(x,u^*) \neq 0$,

then the adjoint λ_x^* must be equal to either its upper or lower bound in (3.9) and the optimal control u^* is found by solving the corresponding minimization (3.8) or maximization (3.7). However, for some states x it may be optimal to choose u^* such that $f(x,u^*) = 0$, and in that case the minimum principle (3.4) gives no information on the adjoint variable λ_x^*. Nevertheless, the condition that the state x tends asymptotically to a terminal state ω implies that $f(x,u^*) = 0$ for optimal u^* if and only if $x = \omega$. Note that such an asymptotic terminal condition is assumed for each initial boundary layer problem in the SPT approximation. Note also that without assuming such an asymptotic terminal condition it may be possible to achieve a lower cost (3.3) with a state trajectory which does not tend to any limit, e.g., a periodic trajectory.

Therefore, let us add to the control problem (3.2) and (3.3) the asymptotic terminal condition

$$\lim_{\tau \to \infty} x(\tau) = \omega \tag{3.10}$$

Then if $u^*(x)$ denotes the optimal feedback control for state x, we have $f(x,u^*(x)) = 0$ if and only if $x = \omega$. In this case, $u^*(x)$ is found from solving the static problem (3.7) when $x < \omega$ and from solving the static problem (3.8) when $x > \omega$. The corresponding optimal adjoint is equal to $m(x)$ if $x < \omega$ and is equal to $M(x)$ if $x > \omega$. This procedure for determining the optimal u and λ_x in terms of x is illustrated in Figure 4.3.1.

To see how the preceding considerations apply to the multi-time scale control problem such as we have illustrated in Figure 4.2.2, let us assume that each of the initial boundary layer control problems in Figure 4.2.2 is one-dimensional, i.e., assume that $n_i = 1$ for each $i = 1,2,\ldots,r$. Then the initial boundary layer problems can be solved as a sequence of static optimization problems as illustrated in Figure 4.3.2. In Appendix 4.2 of this chapter we show that this static optimization problem has a solution under very general conditions.

Although we avoid having to solve any TPBVP's or dynamic trajectory optimization problems for the initial boundary layer problems, the complete separation of time scales in the boundary layer problems may result in a static optimization problem which is no less difficult to solve than the corresponding dynamic optimization problem. In particular, note that as before the cost criterion and the terminal condition at each level depend on the computations of

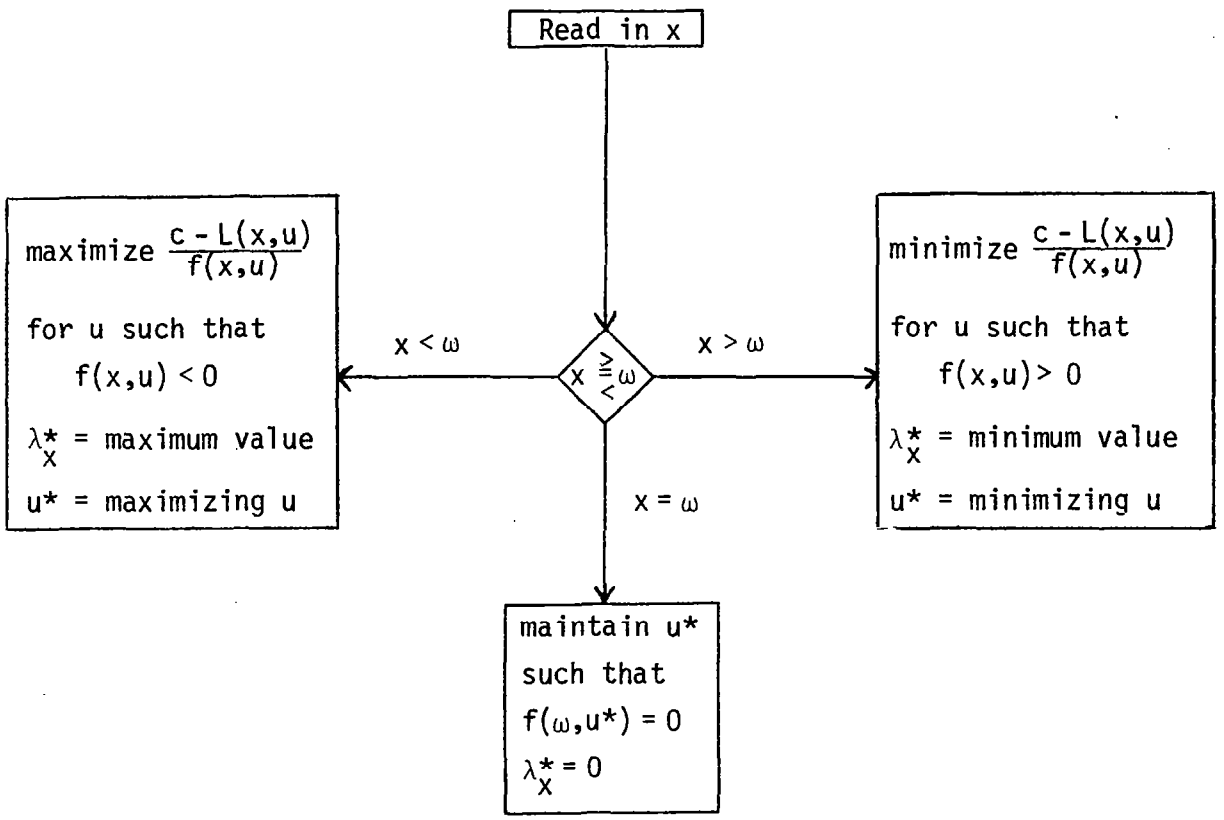


Figure 4.3.1.-Feedback Control Algorithm for One-Dimensional State.

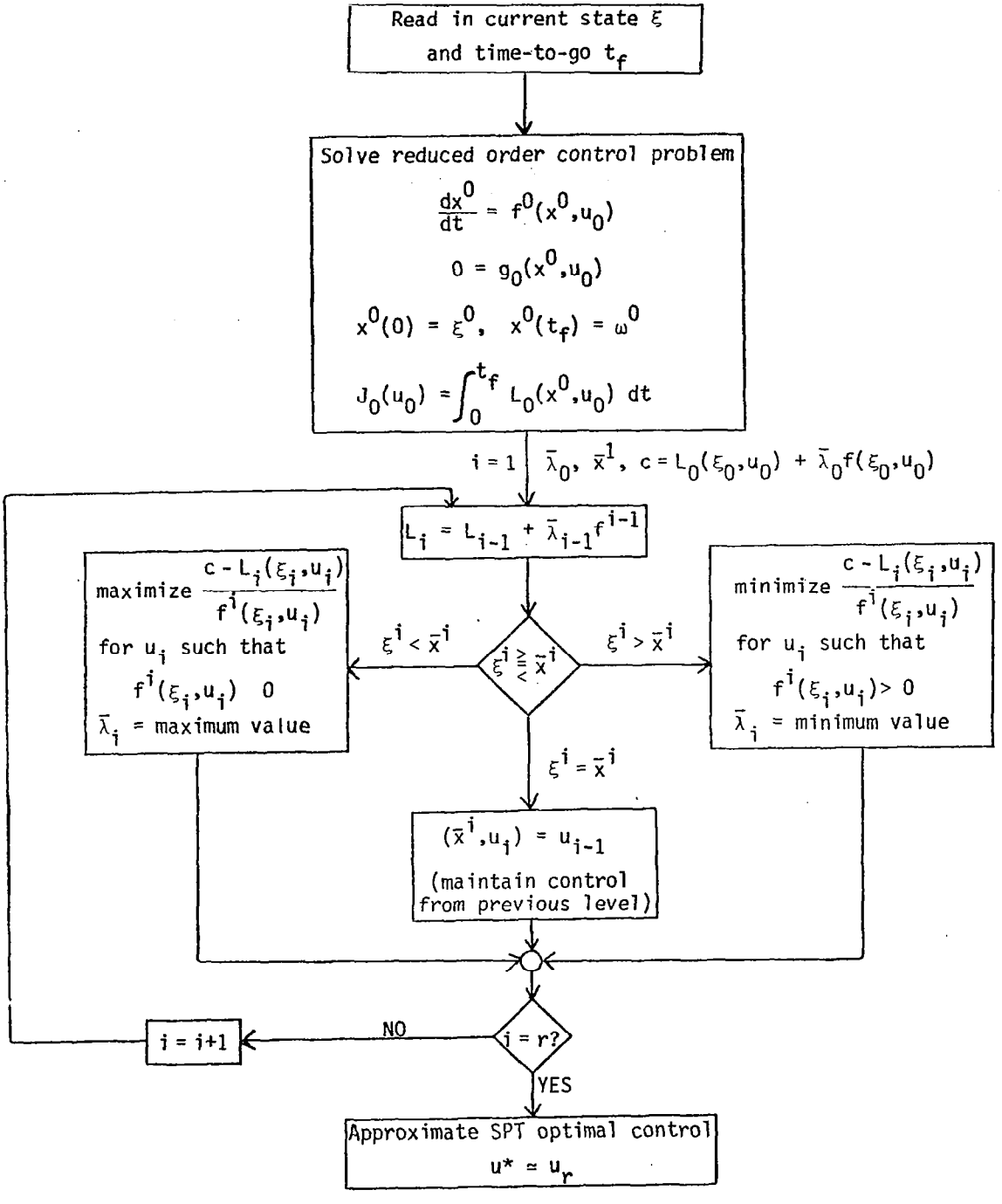


Figure 4.3.2.-Feedback Control Algorithm for Complete Time Scale Separation.

the previous level. Thus, the cost criterion and the terminal condition as functions of ξ_i become more and more complex as i increases. In the highest level, i = r, we will have a complex nonlinear programming problem whose criterion function L_r depends on ξ, the current state, and cannot be determined until the previous r-1 initial boundary layer problems have been solved.

4.3.3 Suboptimal Approximation of the Reduced Order Solution

A second possibility for increasing the efficiency of the SPT algorithm is the use of suboptimal approximations for the reduced order solution. As shown in Figure 4.2.2, the reduced order control problem is the first step in the SPT algorithm for approximating the optimal control law. By approximating the solution of the reduced order problem, we may be able to solve a reasonably large dimensional reduced order problem in an efficient manner and leave a small dimensional initial boundary layer problem to solve in the next step of the SPT algorithm. In this way, we may reduce the computation time for the whole algorithm and at the same time, because we can consider a larger dimensional reduced order problem rather than an artificially small dimensional reduced order problem, we may also improve the accuracy of the SPT approximation.

For simplicity let us only consider the suboptimal approximation of the reduced order solution in the case of two time scales as in Subsection 4.2.2. There is no difficulty in applying the same procedure to the multi-time scale case. In order to go from the reduced order control problem to the initial boundary layer problem (see Figure 4.2.1), it is necessary to have the adjoint $\bar{\lambda}_x^*$ of the reduced order state variable $\bar{x}$. Not only does this adjoint define the cost criterion for the initial boundary layer problem, it also determines the asymptotic terminal condition $\bar{y}^*$ of the initial boundary layer problem as a solution of the minimization:

$$(3.11) \qquad \min \{L(\xi,\bar{y},\bar{u}) + \bar{\lambda}_x^* f(\xi,\bar{y},\bar{u}): \quad g(\xi,\bar{y},\bar{u}) = 0\}$$

Unfortunately, we may not have the adjoint $\bar{\lambda}_x^*$ immediately available. For example, we might be given a suboptimal feedback control $\bar{u}_s(\bar{x})$ and $\bar{y}_s(\bar{x})$ derived from heuristic considerations or from previous experience and have no corresponding adjoint. Moreover, if the reduced order control law is not optimal, it is not clear what the adjoint of the suboptimal reduced order state should be.

To resolve these difficulties we must consider the behavior of the initial boundary layer solution. As we noted previously, this problem is an infinite horizon control problem with the infinite time cost criterion defined in (2.74). Assuming that the boundary layer state $\hat{y}$ has an asymptotic limit as in (2.73), then it is not hard to see that this limit must be the first component $\bar{y}^*$ of the solution of the minimization problem in (3.11). That is, $\bar{y}^*$ is the optimum steady state control for the reduced order problem with cost (2.74). Thus, we should choose the adjoint $\bar{\lambda}_x^*$ so that the solution of (3.11) gives a reasonable suboptimal control for the reduced order control problem.

For example, if we are given the suboptimal control $\bar{u}_s, \bar{y}_s$ for the reduced order problem and we wish to find an adjoint which gives back this particular suboptimal control, then we must find $\bar{\lambda}_x^*$ such that $\bar{u}_s, \bar{y}_s$ is the solution of (3.11). For example, if $\bar{u}_s$ and $\bar{y}_s$ do not occur on a constraint boundary of $\bar{y}$ or $\bar{u}$, then we may determine the adjoint $\bar{\lambda}_x^*$ by solving the simultaneous linear equations

$$(3.12) \qquad \frac{\partial L}{\partial y}(\xi,\bar{y}_s,\bar{u}_s) + \bar{\lambda}_x^* \frac{\partial f}{\partial y}(\xi,\bar{y}_s,\bar{u}_s) + \bar{\lambda}_y^* \frac{\partial g}{\partial y}(\xi,\bar{y}_s,\bar{u}_s) = 0$$

$$(3.13) \qquad \frac{\partial L}{\partial u}(\xi,\bar{y}_s,\bar{u}_s) + \bar{\lambda}_x^* \frac{\partial f}{\partial u}(\xi,\bar{y}_s,\bar{u}_s) + \bar{\lambda}_y^* \frac{\partial g}{\partial u}(\xi,\bar{y}_s,\bar{u}_s) = 0$$

Note that unless the dimension of the x state is equal to the dimension of the u control, there will not be the same number of equations as unknown adjoint variables in (3.12) and (3.13). Thus, in general we must expect that the simultaneous equations in (3.12) and (3.13) will have either no solutions or infinitely many solutions. If the equations (3.12), (3.13) have no solution, then we cannot obtain $\bar{\lambda}_x^*$ with this method and we must find some other way of choosing an appropriate adjoint $\bar{\lambda}_x^*$ for the reduced order control problem.

Instead of trying to find an adjoint $\bar{\lambda}_x^*$ which will yield a given reduced order control $\bar{y}_s, \bar{u}_s$ as the solution of (3.11), let us find an adjoint which will yield a <u>better</u> reduced order control than the control $\bar{y}_s, \bar{u}_s$. It is clear that such a method would be preferable to the first method described above, but it is not clear how we can find the desired adjoint. Fortunately, Bellman's (1954, 1957, 1961) technique of monotone approximation provides exactly the solution we are seeking.

Since monotone approximation is an application of the ideas of dynamic

programming, we start by briefly describing the role of dynamic programming in optimal control. The reader should refer to Bellman (1957) for details. Suppose that we have an autonomous dynamic system

$$\frac{dx}{dt} = f(x,u) \tag{3.14}$$

with the cost criterion

$$J(u) = \int_0^T L(x,u)\, dt + C(x(T)) \tag{3.15}$$

and a fixed terminal time T. Let the state have the following initial condition

$$x(0) = \xi \tag{3.16}$$

Note that for mathematical convenience we have fixed the terminal time T and we have omitted a terminal condition in favor of the terminal cost term C in the cost criterion (3.15).

Let $V^*(\xi,T)$ denote the minimum possible cost (3.15) over all admissible controls u. That is, let

$$V^*(\xi,T) = \min_u J(u) \tag{3.17}$$

We can interpret $V^*(\xi,T)$ as the minimum cost-to-go from the state ξ with time-to-go T. The minimum cost-to-go V^* satisfies the following functional equation:

$$\frac{\partial V^*}{\partial T} = \min_u \{L(\xi,u) + \frac{\partial V^*}{\partial \xi} f(\xi,u)\} \tag{3.18}$$

with the boundary condition

$$V^*(\xi,0) = C(\xi) \tag{3.19}$$

If $u^*(\xi,T)$ is an optimal control to use at state ξ with time-to-go T, then (3.18) implies

$$\frac{\partial V^*}{\partial T} = L(\xi,u^*(\xi,T)) + \frac{\partial V^*}{\partial \xi} f(\xi,u^*(\xi,T)) \tag{3.20}$$

Monotone approximation uses the boundary condition (3.19) and the linear equation (3.20) in V* to find a cost-to-go function V_s corresponding to a given suboptimal control law u_s. Thus, suppose that V_s satisfies

$$(3.21) \qquad \frac{\partial V_s}{\partial T} = L(\xi,u_s(\xi,T)) + \frac{\partial V_s}{\partial \xi} f(\xi,u_s(\xi,T))$$

with the boundary condition (3.19). Then if we generate a new suboptimal control u_s' by minimizing the function

$$(3.22) \qquad L(\xi,u) + \frac{\partial V_s}{\partial \xi} f(\xi,u)$$

with respect to u, the new cost associated with this new control law u_s' will be less than the cost for the control law u_s. By repeating this process one obtains a sequence of cost-to-go functions which decrease monotonically to the optimal cost-to-go V*.

To obtain an appropriate adjoint to use in the initial boundary layer problem, we only need to apply the monotone approximation method once. Thus, given a suboptimal control law u_s, we solve (3.21), (3.19) to obtain $V_s(\xi,T)$. The adjoint corresponding to the current state ξ with time-to-go T is then the partial derivative $\frac{\partial V_s}{\partial \xi}$. Thus, we choose as an adjoint the variation with respect to the initial state or the cost-to-go from that state.

To summarize, the monotone approximate method for obtaining the reduced order adjoint consists of the following steps:

(i) Solve the linear partial differential equation

$$(3.23) \qquad \frac{\partial V}{\partial T} = L(\xi,\bar{y}_s(\xi,T),\bar{u}_s(\xi,T)) + \frac{\partial V}{\partial \xi} f(\xi,\bar{y}_s(\xi,T),\bar{u}_s(\xi,T))$$

with an "appropriate" terminal condition of the form

$$(3.24) \qquad V(\xi,0) = C(\xi)$$

If a terminal condition $\bar{x}_f$ is specified for the reduced order control problem rather than a terminal cost C as above, then we may introduce a fictitious cost such that $C(\bar{x}) = 0$ for $\bar{x} = \bar{x}_f$ and $C(\bar{x}) = +\infty$. A finite cost in (3.24) would be more

tractable mathematically and computationally, and perhaps more realistic than a fixed terminal condition. After solving (3.23), (3.24), obtain the adjoint $\bar{\lambda}_x^*$ from

$$(3.25) \qquad \bar{\lambda}_x(\xi,T) = \frac{\partial V}{\partial \xi}(\xi,T)$$

(ii) Using the adjoint calculated in step (i), minimize the function

$$(3.26) \qquad L(\xi,y,u) + \bar{\lambda}_x(\xi,T)f(\xi,\bar{y},\bar{u})$$

with respect to the reduced order controls $\bar{u}$ and $\bar{y}$. The solution of this minimization problem provides the necessary asymptotic terminal condition for the initial boundary layer control problem.

To implement this method for finding the reduced order adjoint, we must obtain an efficient solution to the partial differential equation in (3.23). Solving partial differential equations is never an easy task, and the computation of the adjoint $\bar{\lambda}_x$ would most likely have to be done off-line. Nevertheless, note that for a given suboptimal control the solution of (3.23) will be much easier than the solution of the nonlinear dynamic programming partial differential equation (3.18) for the optimal cost-to-go. Moreover, a number of techniques are available for solving and approximating the linear problem.

In Figure 4.3.3 we have illustrated the basic procedure of using a suboptimal reduced order control. Note that in the multi-time scale case we may use this suboptimal strategy at any level of the SPT algorithm. Thus, we may compute a suboptimal solution of the $i^{\underline{th}}$ initial boundary layer control problem and use one of the above methods to compute an adjoint variable to use in defining the cost criterion for the $(i+1)^{\underline{th}}$ problem.

4.3.4 Linearization of the Boundary Layer Problems

A third possibility for increasing the efficiency of the SPT algorithm is the linearization of the initial boundary layer control problem around the nominal provided by the first component of the pseudocontrol of the reduced order solution. Implicitly, this approximation assumes that the "fast" boundary layer variables are approximately in equilibrium; that is, equation (2.54) is approximately correct. As we have done in the previous two subsections, we

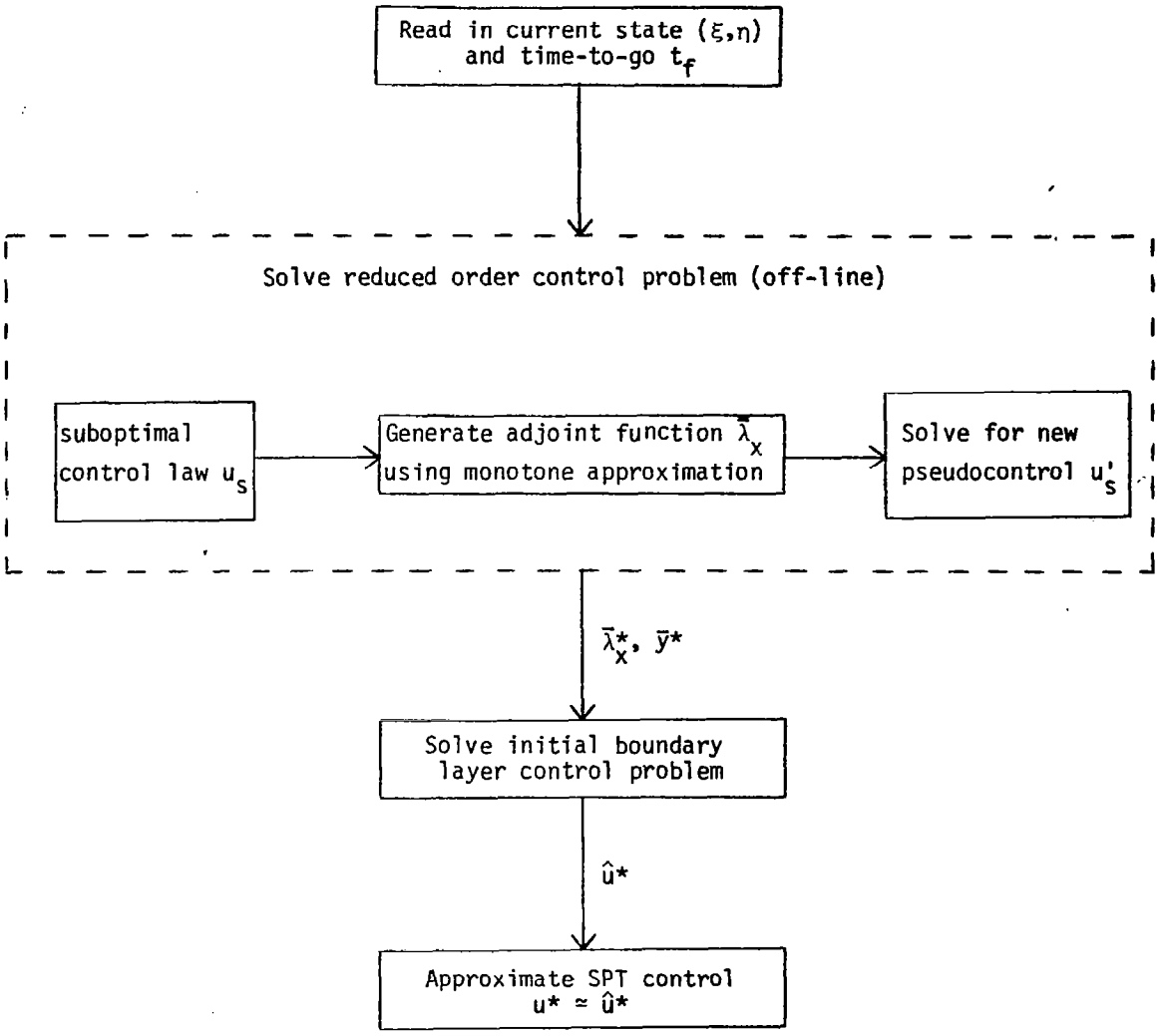


Figure 4.3.3.-SPT Algorithm Using Suboptimal Reduced Order Solution.

will first discuss the two time scale case in detail and then briefly discuss the multi-time case.

Consider the two time scale problem in Subsection 4.2.2 and let the current x state be fixed at ξ. Let $\bar{\lambda}_x^*$, $\bar{y}^*$, and $\bar{u}^*$ be the corresponding optimal adjoint $\bar{\lambda}_x$ and pseudocontrols $\bar{y}$, $\bar{u}$ to use for this value of the x state in the reduced order control problem. We are going to linearize the boundary layer control problem (2.71) - (2.74) around these optimal reduced order values. In the following derivation let us denote quantities evaluated at the optimal reduced order solution by a bar, e.g., $\overline{\frac{\partial g}{\partial y}}$. Let δy denote $\hat{y} - \bar{y}^*$ and let δu denote $\hat{u} - \bar{u}^*$. Expanding (2.71) to first order gives

$$(3.27) \qquad \delta y' = \overline{\frac{\partial g}{\partial y}}\,\delta y + \overline{\frac{\partial g}{\partial u}}\,\delta u$$

where the derivative with respect to τ is denoted by a prime, i.e., $(\ )' = \frac{d(\ )}{d\tau}$. Let us define matrices A and B as

$$(3.28) \qquad A = \frac{\partial g}{\partial y}(\xi, \bar{y}^*, \bar{u}^*)$$

$$(3.29) \qquad B = \frac{\partial g}{\partial u}(\xi, \bar{y}^*, \bar{u}^*)$$

Then (3.27) represents the time invariant linear system

$$(3.30) \qquad \delta y' = A\delta y + B\delta u$$

To obtain the quadratic cost criterion for the linearized problem we first define the Hamiltonian for the boundary layer problem, that is,

$$(3.31) \qquad H(\hat{y},\hat{u}) = L(\xi,\hat{y},\hat{u}) + \bar{\lambda}_x^* f(\xi,\hat{y},\hat{u}) + \hat{\lambda}_y g(\xi,\hat{y},\hat{u})$$

From Bryson and Ho (1975) we see that the cost criterion for the linearized boundary layer problem is defined in terms of the following three matrices which consist of the second order derivatives of this Hamiltonian.

$$(3.32) \qquad Q = \overline{\frac{\partial^2 H}{\partial y^2}}$$

$$(3.33) \qquad R = \overline{\frac{\partial^2 H}{\partial y \partial u}}$$

$$(3.34) \qquad S = \overline{\frac{\partial^2 H}{\partial u^2}}$$

In the expressions (3.32) - (3.34) the bar denotes that the expression is evaluated at the reduced order nominal value. Thus, for example, we have

$$(3.35) \qquad \overline{\frac{\partial^2 H}{\partial y^2}} = \frac{\partial^2 L}{\partial y^2}(\xi,\bar{y},\bar{u}) + \bar{\lambda}_x \frac{\partial^2 f}{\partial y^2}(\xi,\bar{y},\bar{u}) + \bar{\lambda}_y \frac{\partial^2 g}{\partial y^2}(\xi,\bar{y},\bar{u})$$

Evidently, we need to calculate the Lagrange multiplier $\bar{\lambda}_y$ in the reduced order problem if we are going to linearize. For example, we can accomplish this by solving the linear equations

$$(3.36) \qquad \overline{\frac{\partial L}{\partial y}} + \bar{\lambda}_x \overline{\frac{\partial f}{\partial y}} + \bar{\lambda}_y \overline{\frac{\partial g}{\partial y}} = 0$$

$$(3.37) \qquad \overline{\frac{\partial L}{\partial u}} + \bar{\lambda}_x \overline{\frac{\partial f}{\partial u}} + \bar{\lambda}_y \overline{\frac{\partial g}{\partial y}} = 0$$

Note that some of these equations will be redundant for the purpose of obtaining $\bar{\lambda}_y$.

Using the expressions (3.32) - (3.34), we can write the cost criterion for the linearized problem as

$$(3.38) \qquad \int_0^\infty \tfrac{1}{2}[\delta y^T Q\, \delta y + 2\delta y^T R\, \delta u + \delta u^T S\, \delta u]\, d\tau$$

Hence, the linearized approximation of the boundary layer control problem gives a time invariant, linear quadratic regulator problem. The feedback control is simply calculated by solving a system of quadratic equations (the so-called algebraic Riccati equation) to obtain a constant gain matrix G. For this problem G is obtained by solving the following equations for G and K simultaneously:

$$(3.39) \qquad G = -[S^{-1}(KB+R)]^T$$

(3.40) $Q + 2RG + G^TSG + KA + KBG = 0$

The optimal control law for δu in terms of δy is then

(3.41) $\delta u = G\delta y$

To summarize, the procedure for finding the linearized boundary layer control is to compute the matrices A, B, Q, R, S from (3.28), (3.29), (3.32), (3.33), (3.34) for the current value of the reduced order state, and then to solve the equations (3.39), (3.40) to obtain the gain G which gives the boundary layer control law in (3.41).

In the multi-time scale case we may carry out the linearization at any level of the SPT algorithm. Thus, it may be appropriate in some cases to linearize the control problem for the $i^{\underline{th}}$ control subproblem around the nominal values proved by the $(i-1)^{\underline{th}}$ control subproblem. Other variations are also possible. For example, we may decide to linearize the boundary layer problem described above around $\bar{y}^*$ only and maintain the nonlinear dependence of the boundary layer problem on the control $\hat{u}$. In any case, the decision whether or not to linearize is determined by the "error" between the actual value of a variable on the $i^{\underline{th}}$ level of the SPT algorithm and its "optimal" value on the $(i-1)^{\underline{th}}$ level. By monitoring this error an algorithm can decide when to switch from a more accurate and also more difficult nonlinear control to a more efficient linear approximation. One such algorithm is illustrated in Figure 4.3.4.

Before concluding this section, let us note some of the advantages and disadvantages of the linearization procedure described in this subsection. *The major disadvantage is that we are using a linear model for a nonlinear system.* In particular, this approximation will be valid only when the boundary layer states are sufficiently close to their "optimal" values calculated in the reduced order solution. The major advantages of linearization are that (1) the linear control is easier to compute than a nonlinear control, and (2) because the linear control is easier to compute, we may treat a larger dimensional boundary layer control problem than otherwise: thus, we may avoid having to introduce multiple time scales artificially in order to obtain a tractable computation. Note that once a linear problem is obtained it is possible to apply the Generalized Multiple Time Scales method (GMS) of Ramnath and Sandri (1969) to improve

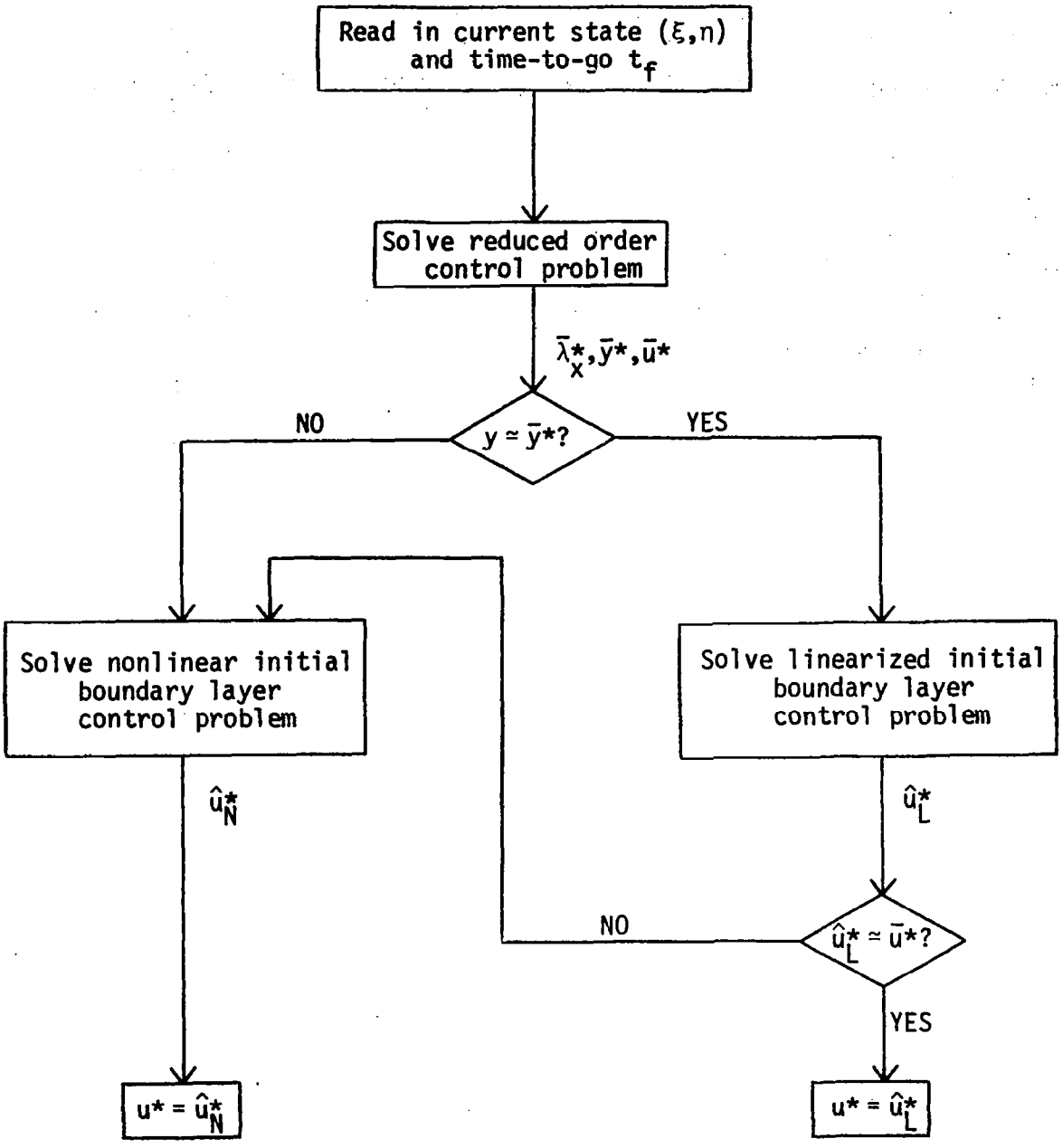


Figure 4.3.4.-SPT Algorithm Using Linearized Boundary Layer Problem.

the efficiency of computation. The time scales for the linearized problem are determined by checking the eigenvalues of the linear system and thus, we can naturally determine the perturbation parameters. This multiple time scaling of the linear system decomposes the control problem and reduces the computation required. In addition, the time scaling improves the numerical accuracy of computation. This latter aspect is most important in view of the numerical sensitivity of the linearization. This sensitivity arises from the presence of tabular, non-analytical functions (aerodynamic coefficients) whose derivatives must be given for the linearization. In Figure 4.3.5 we have summarized the principal advantages and disadvantages of the three strategies we have discussed in this section for making the SPT algorithm more efficient.

4.4 Accuracy of the SPT Approximation

4.4.1 State Space Dependence of the Accuracy of the SPT Approximation

The SPT algorithm discussed in the previous sections gives a control law $u(\xi,T)$ for the control problem (2.1) - (2.4) in terms of the current state ξ and the time-to-go $T = t_f - t_0$. Let $J(u;\xi,T)$ denote the cost-to-go starting at the state ξ with time-to-go T and using the control law u. If $V^*(\xi,T)$ is the minimum cost-to-go from the state ξ with time-to-go T, then the expression $e(u;\xi,T)$ given by

$$(4.1) \qquad e(u;\xi,T) \triangleq J(u;\xi,T) - V^*(\xi,T)$$

defines a measure of the accuracy of the control law u compared to the optimal control. Since V^* is the optimal cost, it is clear that e is always nonnegative. However, as the initial state ξ and the time-to-go T vary, we may expect that the error e will vary also, being greater in some regions and smaller in others.

Let $\hat{u}$ specifically denote the SPT approximate optimal control law. For any fixed initial state $\xi \neq \omega$ and time-to-go T the error $e(\hat{u};\xi,T)$ depends on the singular perturbation parameter ε. As ε decreases to 0 we expect the error to approach 0. However, the error does not approach 0 uniformly in ξ and T. That is, for a fixed perturbation parameter ε, the error may vary greatly for different values of ξ and T. *In this section we would like to show that the SPT approximation error increases as the initial state ξ approaches the terminal state ω. That is, as one nears the terminal target state, the SPT approximation breaks*

SUMMARY OF PRINCIPAL ADVANTAGES AND DISADVANTAGES OF COMPLETE TIME SCALE SEPARATION, SUBOPTIMAL REDUCED ORDER SOLUTION, AND LINEARIZATION OF BOUNDARY LAYER CONTROL PROBLEM

1. Complete time scale separation (see Subsection 4.3.2)

   Advantage: Reduces dynamic trajectory optimization problem to static optimization problem for control law.

   Disadvantage: Introduces multiple time scales artificially and will be inaccurate if some state variables are highly coupled; the sequence of static optimization problems may be as difficult to solve as the original dynamic problems for a large dimensional problem where many time scales are necessary.

2. Suboptimal Reduced Order Solution (see Subsection 4.3.3)

   Advantage: Allows one to avoid solving the reduced order problem exactly; thus, it is possible to treat larger dimension reduced order problems.

   Disadvantage: The suboptimal adjoint function must be computed and approximated off-line, although this should be easier to do than to compute the optimal adjoint.

3. Linearization of the Boundary Layer Problem (see Subsection 4.3.4)

   Advantage: Linearized controls can be computed efficiently and larger dimension boundary layer problems can be treated.

   Disadvantage: Linearization is valid only when the boundary layer state is near its optimal reduced order value; thus, linearization essentially gives only a control to track the reduced order solution, and it will be able to do this only as long as the pseudocontrol for the reduced order problem does not jump discontinuously.

Figure 4.3.5.-Comparison of Advantages and Disadvantages of Complete Time Scale Separation, Suboptimal Reduced Order Solution and Linearized Boundary Layer Approximation.

down. Before discussing this breakdown in more detail, however, let us note that the opposite phenomenon is true when the initial state is far from the terminal target, at least in some cases; that is, *the SPT approximation improves as the initial state ξ increases its distance from the terminal state ω.*

As we will see in the next chapter, the aircraft dynamical system has the general form

$$\frac{dx}{dt} = f(y,u) \tag{4.2}$$

$$\frac{dy}{dt} = g(y,u) \tag{4.3}$$

where the right hand side of (4.2) and (4.3) do not depend on x. In the aircraft problem, (4.1) corresponds to the equations for the horizontal position coordinates of the aircraft. The variables y are such that either through direct state constraints or by definition (as in the case of angles) each component of y has a maximum value. In the aircraft example y includes height, velocity, heading angle and so on. The x variables, on the other hand, have a potentially infinite range. If we scale the variables x and y such that the scaled quantities x' and y' attain a maximum variation of order of magnitude 1, and if we scale the time variable t so that the scaled time t' varies from 0 to 1, then the original dynamic system (4.2), (4.3) is transformed to

$$\frac{dx'}{dt'} = f'(y',u) \tag{4.4}$$

$$\varepsilon\frac{dy'}{dt'} = g'(y',u) \tag{4.5}$$

where the parameter ε is essentially the ratio of the maximum variation of y to the initial distance from x to its target state x_f. Moreover, the functions f' and g' in the aircraft example do not depend on the parameter ε. *Thus, in the minimum time problem for the aircraft example there is at least one natural singular perturbation parameter, and this parameter is proportional to the distance from the target.* Since we expect that the SPT approximation is best when ε is small, we expect that the SPT approximation for the aircraft will be good when the range is sufficiently long, but that the SPT approximation will break down near the target when the range is short. Note that for the example

we have considered, the SPT approximation starts to break down when the scale parameter ε in (4.4), (4.5) increases above .07. This corresponds to target distances less than 163 to 176 Km.

4.4.2 Inaccuracy of the SPT Approximation Near the Terminal Target State

It is the purpose of this subsection to show that one must be especially careful when using SPT approximations to obtain closed loop, control law approximations rather than open loop, control trajectory approximations. At the terminal time the SPT control law will not be the asymptotic approximation of the actual optimal feedback control law. To illustrate what goes wrong we will consider the following simple linear quadratic control problem.

$$\varepsilon\frac{dx}{dt} = u \tag{4.6}$$

with the initial and terminal conditions $x(0) = \xi$ and $x(T) = 1$, respectively. The cost criterion for the problem is

$$J(u) = \int_0^T \tfrac{1}{2}(x^2 + u^2)\, dt \tag{4.7}$$

The exact optimal control law is easily found to be

$$u(\xi,T) = \frac{2e^{-T/\varepsilon} - x(1+e^{-2T/\varepsilon})}{(1-e^{-2T/\varepsilon})} \tag{4.8}$$

The $0^{\underline{th}}$ order SPT approximation, the first term of (2.6), is also easily found to be

$$\hat{\tilde{u}}(\xi,T) = -x + e^{-T/\varepsilon} \tag{4.9}$$

Note that we have written "$\hat{\tilde{u}}$" in (4.9) to indicate that both the $\hat{u}$ and $\tilde{u}$ (as well as $\bar{u}$) terms are accounted for.

To see how the SPT approximate control law in (4.9) differs from the exact optimal feedback control law given in (4.8), we approximate the latter to first order. First consider the region in which the time-to-go T satisfies $T \gg \varepsilon$. The optimal feedback law (4.8) is then approximately, to order $O(e^{-T/\varepsilon})$,

(4.12) $\hat{\tilde{u}}(\xi,T) \simeq -(\xi-1)$

In this control problem (4.6), (4.7) one desires to reach the terminal target $x = 1$ by a fixed terminal time, say $t_f = 1$. Then as the current time t nears the terminal time t_f the optimal control law (4.8) takes this into account through the time-to-go $T = t_f - t$ becoming small. As the time-to-go T becomes smaller, the optimal control law places more emphasis on hitting the target exactly at the right final time t_f and less importance on reducing the quadratic cost. The SPT control law (4.9), on the other hand, resembles a linear regulator control around the terminal state $x = 1$. Although such a control would steer the state toward $x = 1$, it would not be able to do so in a finite amount of time. Hence, at the terminal time t_f the SPT controlled state will miss the desired terminal state. To see more clearly what trajectory the SPT control produces, let us integrate (4.6) with $u = \hat{\tilde{u}}$. Then we find that if ξ is the initial state with time-to-go T, the final state $x(t_f)$ using SPT control will be

(4.13) $$x(t_f) = e^{-T/\varepsilon}\xi + \frac{\varepsilon}{2}\left(1 - e^{-2T/\varepsilon}\right)$$

If $x = 1$ is the desired terminal state, then the error between this and the actual terminal state (4.13) is given by

(4.14) $$1 - x(t_f) = 1 - \frac{\varepsilon}{2} + \frac{\varepsilon}{2} e^{-2T/\varepsilon} - e^{-T/\varepsilon}\xi$$

Thus, if the initial state ξ is such that $e^{-T/\varepsilon}\xi$ is small and the initial time-to-go T is large compared to ε, then the final error at the terminal time t_f will be approximately

(4.15) $$1 - x(t_f) \simeq 1 - \frac{\varepsilon}{2}$$

In fact, for any initial condition ξ, if the initial time-to-go is large enough, then (4.15) will be true.

The situation is worse than indicated by the approximation of u given in (4.12)! If we use the SPT control all along and the state x approaches too closely to its optimal reduced order value $\bar{x}^* = 0$, then x will still be close to the reduced order value at the final time. To be exact, if at any time we ever

have $x < \frac{\varepsilon}{2}$, then $x(t_f) < \varepsilon$.

A similar phenomenon occurs for more complex systems than the simple one we have examined here. Essentially, the SPT control follows the reduced order solution and is unable to anticipate discontinuities in the reduced order solution in order to cross the jump of the discontinuity in the required amount of time. In this example the reduced order value of x was 0 and the desired terminal value was 1. Although the SPT control law begins to steer x away from the reduced order value 0, it does not do it fast enough to meet the terminal value and the resulting error (4.15) is fairly large.

The trouble we have described is not limited to the terminal region: so-called internal boundary layers exhibit a similar behavior. To describe the situation, let us refer to the two time scale problem of Subsection 4.2.2. Internal boundary layers occur when the reduced order control problem gives a pseudocontrol $\bar{y}^*(\xi,T)$ with discontinuities for some values of current x-state ξ and time-to-go T. Such discontinuities indicate that the real y-state, which must be continuous, changes very rapidly. This rapid change is approximated by a boundary layer and this requires us to modify our basic time scale approximation (2.38) of the optimal control u^* for the full system as follows:

$$(4.16) \qquad u^* \simeq \sum_{k=0}^{\infty} \{[\hat{u}_k(\tau) - \bar{u}_k(t_0)] + [\ddot{u}_k(\rho) - \bar{u}_k(t_i)] + \bar{u}_k(t) + [\tilde{u}_k(\sigma) - \bar{u}_k(t_f)]\}\, \varepsilon^k$$

where t_i is the time at which the reduced order solution exhibits a discontinuity in the pseudocontrol $\bar{y}^*$ and ρ is the fast time scale for the internal boundary layer given by $\rho = \frac{t-t_i}{\varepsilon}$.

The argument we made in Subsection 4.2.1 concerning the terminal boundary layer also holds for the internal boundary layer--as long as the time t is not close to the time t_i. We may neglect the contributions $\ddot{u}_k$ from the internal boundary layer. Near to the internal boundary layer time the SPT approximation breaks down in the same way that it did at the terminal time. Essentially, the SPT control law does not anticipate the sudden rapid change in the y variable. An optimal control would begin steering y across the discontinuity before the internal boundary layer time t_i was reached. The SPT control, on the other

hand, does not do this until t_i is reached, and then it steers y to the reduced order value $\bar{y}^*$ after t_i.

At the present time we do not have a solution to this fundamental problem of using SPT control law approximations near the terminal target or at internal boundary layers, but clearly the problem is essential for any application of SPT methods to obtain feedback control laws. At the very least, it will be necessary to identify the regions where the SPT feedback law breaks down so that one may switch from the SPT control to a better control law. For obtaining the control law near the terminal target, it may be possible to develop useful approximations different from the SPT approximation. For example, linearizing some of the state variables around their terminal values may prove useful. However, whatever method of approximation is used near the terminal target, the important problem to solve will be to determine how to match this approximation with the SPT approximation--in other words, to determine when and how to switch from one type of control to another.

APPENDIX 4.1
MULTI-TIME SCALE SPT FORMULATION AND CALCULATION

In this appendix we present the SPT formulation and calculation of the feedback law in more detail than in Subsection 4.2.3. Throughout this appendix we will maintain the same notation as we did in that subsection. In addition, let us define x_i as

$$(1.1) \qquad x_i = (x^1,\ldots,x^i)$$

just as we defined ξ^i in (4.2.79). Likewise, define f_i by

$$(1.2) \qquad f_i(x,u) = (f^1(x,u),\ldots,f^i(x,u))$$

The SPT method seeks to approximate the control system in equations (4.2.1) through (4.2.4) by finding an asymptotic expansion to the singularly perturbed system

$$(1.3) \qquad \varepsilon_i \frac{dx^i}{dt} = f^i(x,u)$$

with the same cost criterion

$$(1.4) \qquad J(u) = \int_{t_0}^{t_f} L(u,x)\, dt$$

the same initial conditions

$$(1.5) \qquad x^i(t_0) = \xi^i$$

and the same terminal conditions

$$(1.6) \qquad x^i(t_f) = \omega^i$$

In (1.3) the parameters ε_i are such that $\varepsilon_0 = 1$ and for each i we have

$$\varepsilon_{i+1}/\varepsilon_i \to 0 \tag{1.7}$$

as $\varepsilon_i \to 0$. In reality, of course, we will be using the approximation for $\varepsilon_i = 1$ for all i. The relation (1.7) represents the assumption that the variables x^i vary on different time scales which become increasingly faster as i increases.

Although the multi-time scale SPT approximation is conceptually no more difficult than the two time scale case, the notational difficulties in the multi-time case become extreme. Therefore, we will try to simplify the problem at an abstract level as much as possible before working on the control problem. Working at the abstract, more general level we can avoid some of the notational obscurities and maintain a modest level of clarity.

The essential idea of the SPT approximation is to approximate a trajectory $z(t)$ for $t_0 \le t \le t_f$, where z represents all the state components, adjoint components and controls. The $0^{\underline{th}}$ order approximation has the general form:

$$z(t) \simeq [\hat{z}(\vec{\tau}) - \bar{z}(t_0)] + \bar{z}(t) + [\tilde{z}(\vec{\sigma}) - \bar{z}(t_f)] \tag{1.8}$$

where the vectors $\vec{\tau}$ and $\vec{\sigma}$ represent the multi-time scales, namely,

$$\vec{\tau} = (\tau_0, \tau_1, \ldots, \tau_r) \tag{1.9}$$

$$\vec{\sigma} = (\sigma_0, \sigma_1, \ldots, \sigma_r) \tag{1.10}$$

where τ_i is defined $\tau_i = \dfrac{t-t_0}{\varepsilon_i}$ and σ_i is defined $\sigma_i = \dfrac{t-t_f}{\varepsilon_i}$. As for the two time scale case, we need only consider the initial boundary layer represented by the term $\hat{z}$ in (1.8). For multiple time scales, the $0^{\underline{th}}$ order approximation of the initial boundary layer term has the form

$$\hat{z}(\vec{\tau}) = \hat{z}_1(\tau_1) + \sum_{j=1}^{r-1} [\hat{z}_{j+1}(\tau_{j+1}) - \hat{z}_j(0)] \tag{1.11}$$

where the terms $\hat{z}_i$ do not denote different components or combinations of components of z as we defined x^i and x_i previously. The $\hat{z}_i$ all have the same dimension and represent the approximations of z on each of the different time

scales τ_i.

The expressions $\hat{z}_i$ must have the asymptotic property that

$$\lim_{\tau_i \to \infty} \hat{z}_i(\tau_i) = \hat{z}_{i-1}(0) \tag{1.12}$$

for each $i \geq 1$. For this reason, the approximation (1.11) of $z(0)$ is given by

$$z(0) \simeq \hat{z}_r(0) \tag{1.12}$$

Compare this to our discussion in Subsection 4.2.1. The approximation (1.11) is analogous to the first term of (4.2.44). Thus, to obtain the SPT approximation to $z(0)$ we need to calculate the value of $\hat{z}_r(0)$, the fastest initial boundary layer approximation at $\tau_r = 0$. To do this, however, requires first computing the slower initial boundary layer approximations.

The dynamic system for the time scale τ_i is obtained by transforming (1.3) to the independent variable τ_i. Thus, we have

$$\frac{dx^j}{d\tau_i} = \frac{\varepsilon_j}{\varepsilon_i} f^j(x,u), \quad 0 \leq j < i \tag{1.14}$$

$$\frac{dx^i}{d\tau_i} = f^i(x,u) \tag{1.15}$$

$$\frac{\varepsilon^i}{\varepsilon^j} \frac{dx^j}{d\tau_i} = f^i(x,u), \quad r \geq j > i \tag{1.16}$$

The $0^{\underline{th}}$ order approximation is obtained by letting $\varepsilon_r \to 0$ and using the relations (1.7). The equations (1.14) - (1.16) then become

$$\frac{dx^j}{d\tau_i} = 0 \tag{1.17}$$

$$\frac{dx^i}{d\tau_i} = f^i(x,u) \tag{1.18}$$

$$0 = f^j(x,u), \quad r \geq j > i \tag{1.19}$$

These three equations can be summarized in the two equations

$$(1.20) \qquad \frac{dx^i}{d\tau_i} = f^i(\xi_{i-1}, x^i, u_i)$$

$$(1.21) \qquad 0 = g_i(\xi_{i-1}, x^i, u_i)$$

The equations (1.20) and (1.21) represent the dynamical equation and equality constraint, respectively, for the $i^{\underline{th}}$ level system operating on the time scale τ_i. The initial conditions for the state x^i in this problem are those given by (1.5). The terminal conditions, however, are determined as a result of the solution of the control problem for the $(i-1)^{\underline{th}}$ level. These terminal conditions are chosen to satisfy the asymptotic relation (1.12). Likewise, the cost criterion for the $i^{\underline{th}}$ level is determined on the $(i-1)^{\underline{th}}$ level by the adjoint for the $(i-1)^{\underline{th}}$ level. The terminal condition is given in (4.2.92) and the cost criterion is given in (4.2.93) of Subsection 4.2.3.

APPENDIX 4.2
THE EXISTENCE OF SOLUTIONS TO THE BOUNDARY LAYER CALCULATION IN THE CASE OF COMPLETE TIME SCALE SEPARATION

In this appendix we consider the SPT algorithm's calculation of the control law for a dynamic system for which we have assumed complete time scale separation. To simplify matters let us first consider the case of two time scales where x and y represent the slow and fast variables respectively. We assume that both are scalar state variables. Let u denote the vector control variable for the problem, and let L(x,y,u) denote the integrand of the cost criterion. The dynamic equations for the system are

$$\frac{dx}{dt} = f(x,y,u) \tag{2.1}$$

$$\frac{dy}{dt} = g(x,y,u) \tag{2.2}$$

As discussed in Subsection 4.3.2, the SPT algorithm calculates the control law for this system in two stages as follows. Let x_f denote the given final value for the x state variable. The first stage of the SPT calculation is to solve

$$\min\left\{\frac{L(x,y,u)}{f(x,y,u)} : g(x,y,u) = 0,\ f(x,y,u) > 0\right\} = -\bar{\lambda}_x^*(x) \tag{2.3}$$

if $x < x_f$, and to solve

$$\max\left\{\frac{L(x,y,u)}{f(x,y,u)} : g(x,y,u) = 0,\ f(x,y,u) < 0\right\} = -\bar{\lambda}_x^*(x) \tag{2.4}$$

if $x > x_f$. Let $\bar{y}^*(x)$ denote the optimal pseudocontrol and let $\bar{u}^*(x)$ denote the optimal control which solve (2.3) and (2.4).

The second stage of the SPT algorithm is to solve

$$\min\left\{\frac{L(x,y,u) + \bar{\lambda}_x^*(x) f(x,y,u)}{g(x,y,u)} : g(x,y,u) > 0\right\} = -\bar{\lambda}_y^*(x,y) \tag{2.5}$$

if $y < \bar{y}^*(x)$ and to solve

$$(2.6) \qquad \max\left\{\frac{L(x,y,u)+\bar{\lambda}_x^*(x)f(x,y,u)}{g(x,y,u)} : g(x,y,u)<0\right\} = -\bar{\lambda}_y^*(x,y)$$

if $y > y^*(x)$. Let $u^*(x,y)$ denote the optimal control which solves (2.5) and (2.6).

The problem is whether or not we can actually obtain a solution to this algorithm, at least in principle. The question arises because it is not at all clear that the minimum or the maximum exist in (2.3), (2.4), (2.5), (2.6). Indeed, given numerical inaccuracies it is possible that numerical optimization algorithms will not converge to a solution *unless care is taken in setting up the problem.* It is the purpose of this appendix to show that under reasonably general circumstances it is possible, at least theoretically, to obtain finite solutions to the minimization and maximization problems above and that we can obtain at least one solution $u^*(x,y)$. Thus, nonconvergence of optimization subroutines within the SPT algorithm must be due to numerical difficulties (which can be repaired) and not due to the lack of a solution to the minimization or maximization problems within the SPT algorithm. Thus, the SPT algorithm will yield a control value for any input state value--at least in theory. This fact was reassuring to know when some stages of the algorithm proved to be sensitive to numerical error.

The first step in our argument is to prove a small lemma. Notice that in this lemma the optimization problem represents an abstracted and boiled down version of one of the stages of the SPT algorithm mentioned above. Thus, to find solutions to the SPT algorithm above, we will only have to apply the lemma to each stage, one after the other. The reader who is unfamiliar with the mathematical analysis required in the statement and proof of this lemma may skip to the theorem and remarks following the theorem. The analysis required here may be found in any introductory book on real analysis such as Rudin (1964). The main result of mathematical analysis which we use is the fact that a continuous function for a compact set K always achieves a finite minimum $f(z_1)$ and a finite maximum $f(z_2)$ for some z_1 and z_2 in K.

Lemma

Let K denote a compact (closed and bounded) set in R^n and let f and L denote continuous functions from R^n into R. Suppose that for all z in K such that $f(z) = 0$ we have

(2.7) $\quad L(z) > 0$

Furthermore, suppose that there is a z_0 in K such that $f(z_0) > 0$.

Then there exists a z^* in K such that $f(z^*) > 0$ and such that for all z in K such that $f(z) > 0$ we have

(2.8) $\quad \frac{L(z)}{f(z)} > \frac{L(z^*)}{f(z^*)}$

That is, z^* minimizes $L(z)/f(z)$ for all z in K such that $f(z) > 0$.

Proof of the Lemma

The proof is rather simple. Essentially we show that (2.7) implies that as $f(z)$ approaches 0, the ratio $L(z)/f(z)$ approaches $+\infty$. Thus, we can show that there is a positive constant δ such that we can restrict ourselves to z such that $f(z) \geq \delta$ is the minimization of $L(z)/f(z)$. Once this is done, standard compactness arguments show that z^* exists.

Let Z denote the set of z in K such that $f(z) = 0$. Note that Z is compact and since L is continuous, L must have a nonzero minimum on Z. That is, there is a constant $\varepsilon > 0$ such that

(2.9) $\quad L(z) \geq \varepsilon$

for all z in Z. Let Z_ρ denote the set of all z in K such that the distance $d(Z,z)$ from z to the set Z satisfies $d(Z,z) < \rho$. The set Z_ρ is relatively open in K, and $K\text{-}Z_\rho$ is compact. Since L is continuous and since Z is compact, (2.9) implies that for sufficiently small ρ, we have

(2.10) $\quad L(z) \geq \frac{\varepsilon}{2}$

for all z in Z_ρ.

Let K_δ denote the set of points z in K such that $f(z) \geq \delta$. Note that K_δ is compact and disjoint from Z. Since Z is also compact, we can choose ρ small enough so that K_δ and Z_ρ are also disjoint.

Choose δ so that $\delta < f(z_0)$ and so that

(2.11) $\quad \frac{\varepsilon}{2\delta} > \frac{L(z_0)}{f(z_0)}$

Next choose ρ so that Z_ρ is disjoint from K_δ and so that (2.10) is satisfied for z in Z_ρ. Then it is clear that for z in Z we have

$$(2.12) \qquad \frac{L(z_0)}{f(z_0)} < \frac{L(z)}{f(z)}$$

Thus, the minimum of $L(z)/f(z)$ must lie in $K\text{-}Z_\rho$. The function $L(z)/f(z)$ is continuous on this compact set $K\text{-}Z_\rho$ and hence, standard theorems imply the existence of a z^* minimizing L/f.

Given the lemma above, it is fairly easy to show that under reasonably general conditions, the SPT algorithm generates a well-defined solution. In the following theorem we note the general conditions under which a solution can be obtained to the SPT algorithm.

Theorem (Existence of Solutions to the SPT Algorithm)

Assume the following are true:

1) For all x there are compact sets, J_x and K_x such that $y \in J_x \subset R^1$ and $u \in K_x \subset R^m$.
2) For all x the functions $(y,u) \to f(x,y,u)$, $(y,u) \to g(x,y,u)$ and $(y,u) \to L(x,y,u)$ are continuous.
3) For all x,y,u such that $f(x,y,u) = 0$ and $g(x,y,u) = 0$ we have $L(x,y,u) > 0$, if $x \neq x_f$.
4) The optimal pseudocontrol $\bar{y}^*(x)$ solution to the first stage of the SPT algorithm is unique for each x.

Then the SPT algorithm has a solution $u^*(x,y)$.

Remarks

Before sketching the proof of this theorem let us comment on the above assumptions (1) - (4). The first assumption in (1) above merely requires that we can restrict y and u to a bounded set in R^1 and R^m respectively for any given x. The second assumption is obvious. The third assumption (3) is naturally satisfied for many cost criteria (e.g., the minimum time and quadratic criteria, but not the minimum fuel criterion). The first three assumptions guarantee a solution to the first stage of the SPT algorithm. The fourth assumption is necessary to guarantee a solution to the second stage of the SPT algorithm. If there is more than one solution $\bar{y}^*(x)$, then the fast boundary

layer problem in the second stage of the SPT algorithm will have more than one possible terminal condition. Such ambiguity would require an investigation going beyond the first order necessary conditions provided by the minimum principle to obtain the actual optimal control $u^*(x,y)$ for the fast boundary layer problem.

Proof of Theorem

The first three assumptions (1) - (3) permit direct application of the lemma to prove the existence of a solution $\bar{y}^*(x)$, $\bar{u}^*(x)$ to the first stage of the SPT algorithm, for $x \neq x_f$. If $x \neq x_f$ and $y \neq \bar{y}^*(x)$, then the fourth assumption (4) implies that for all u such that $g(x,y,u) = 0$ we have

$$(2.13) \qquad L(x,y,u) + \lambda_x f(x,y,u) > 0$$

From (2.13) we can again apply the lemma directly to prove the existence of a solution $u^*(x,y)$ to the second stage of the SPT algorithm. Note that $u^*(x,y)$ need not be unique. ///

For more than two time scales with complete time scale separation, the existence results are similar. The essential assumption, besides the obvious continuity and compactness assumptions, is the positivity of the original cost criterion when the slowest state variable is not at its terminal value (i.e., we also assume that the generalization of assumption (3) is true), and in addition, the optimal pseudocontrol for the next faster state variable is unique at each stage. For example, if there were a state variable z which was faster than y, then we would have to assume that the pseudocontrol value for z calculated at the second stage (y boundary layer problem) was unique. Note again that this assumption is necessary even to make sense of the SPT algorithm.

(1.15) $T_{max} = T_{max}(h,E)$

(1.16) $L = L_{\alpha}(h,E)\alpha$

(1.17) $V_{max} = V_{max}(h)$

Note that in the energy state formulation the velocity V is simply shorthand for the function of E and h given by

(1.18) $V = \sqrt{2g(E-h)}$

Note also that E is a specific energy measured in units of height. The constraint bounds α_s and σ_{max} are constants in (1.9) and (1.11) respectively. The constraint (1.13) is present only to guarantee that the square root in (1.18) is well-defined. In a model which includes both E and h as state variables this constraint will not be necessary, since the dynamic equations will never allow the height h to exceed the energy E. However, in reduced order models which include only E the constraint will be necessary since h will be a free control.

The control problem is to steer the system (1.3) - (1.8) from an initial state $(x_i,y_i,E_i,\beta_i,h_i,\gamma_i)$ at t_0 to a final state $(x_f,y_f,E_f,\beta_f,h_f,\gamma_f)$ in minimum time t_f-t_0. Thus, the integral cost criterion for this problem is the same as (3.2.6), namely

(1.19) $$J(\alpha,\sigma,u) = \int_{t_0}^{t_f} 1 \, dt$$

Since the final time t_f is not fixed, the Hamiltonian for this problem is identically 0 along an optimal trajectory. Note that the Hamiltonian is given by

(1.20) $$H = 1 + \lambda_x V \cos\beta\cos\gamma + \lambda_y V \sin\beta\cos\gamma + \lambda_E \frac{V(uT_{max}\cos\alpha - D)}{mg}$$
$$+ \lambda_\beta \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} + \lambda_h V \sin\gamma + \lambda_\gamma \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV}$$

To approximate the optimal control law for this minimum time problem we will follow the procedure outlined in Section 4.2 of solving a sequence of smaller

dimensional control problems corresponding to different time scales for the problem.

At this point we will indicate how the time scale separation was chosen. Previous work such as Kelley (1971a,b, 1973c), Kelley and Lefton (1972a) and Parsons (1972), which considered three dimensional maneuvers, and hence considered turns, indicated that the basic time scale separation consisted of the successively faster groups (x,y), (E,β) and (h,γ). Generally, β was considered faster than E and γ was considered faster than h, although Kelley (1973c) noted that E might be considered faster than β in some situations. As we have discussed in Section 4.4, the accuracy of a feedback control calculated with a particular time scale separation depends on the region of state space in which the control is applied. It seems clear from examples that no one time scale separation will be accurate for all regions of state space. Indeed, near the terminal target all time scale separations are inaccurate, i.e., the SPT assumption of any time scale separation is invalid near the terminal target.

Although there is as yet no systematic theory to determine the proper time scale separation for a given region of state space, we present the following observations for guidance. The time scale separation for an optimization problem seems to depend on two factors: (1) the sensitivity of the immediate cost with respect to the current state variables and (2) the sensitivity of the long term cost with respect to the current state variables. It appears generally that the most sensitive states should be considered slower than the least sensitive states. In addition, it appears that the sensitivity of the long range cost is important when the target is near (short term problem) and that the sensitivity of the immediate cost is important when the target is far (long range problem). It is important to note that sensitivity analysis of the two kinds of cost, immediate and long-range, can result in different choices of time scales. The right choice depends on the proximity of the target.

The sensitivity of immediate cost is determined mainly by the relative magnitude of the time derivatives, at least in the minimum time optimization problem. The most sensitive states (with respect to immediate cost) are the ones with the smallest derivatives and hence the slowest in terms of actual time variation.

We carried out a sensitivity analysis for the aircraft model presented in Chapter 1 as follows. We first scaled all state variables and derivatives so

that the state variables and time-to-go were of order one (the maximum variation of each variable was one). That is, we let $\bar{x} = \frac{x}{R_s}$, $\bar{y} = \frac{y}{R_s}$ where R_s = range-to-go, $\bar{V} = \frac{V}{V_s}$ where V_s = maximum speed, $\bar{h} = \frac{h}{h_s}$ where h_s = the cruise altitude, $\bar{E} = \frac{E}{E_s}$ where E_s = cruise energy, $\bar{t} = \frac{t}{t_s}$ where t_s = minimum time-to-go, namely R_s/V_s. Since β and γ are angles and already vary on the order of one, we left them unscaled. Similarly, we scaled lift, drag and thrust in terms of their maximum values. With this scaling we obtained the following equations:

$$(1.21) \qquad \frac{d\bar{x}}{d\bar{t}} = \bar{V}\cos\beta\cos\gamma$$

$$(1.22) \qquad \frac{d\bar{y}}{dt} = \bar{V}\sin\beta\cos\gamma$$

$$(1.23) \qquad \varepsilon\frac{d\bar{E}}{dt} = \delta_E\bar{V}\bar{F}_{||}$$

$$(1.24) \qquad \varepsilon\frac{d\bar{h}}{d\bar{t}} = \delta_V\bar{V}\sin\gamma$$

$$(1.25) \qquad \varepsilon\frac{d\gamma}{d\bar{t}} = \frac{\bar{F}_{||}}{\bar{V}} - \delta_W\frac{\cos\gamma}{\bar{V}}$$

$$(1.26) \qquad \varepsilon\frac{d\beta}{d\bar{t}} = \frac{\bar{F}_{\perp}\sin\sigma}{\bar{V}\sin\gamma}$$

where $\bar{F}_{||}$ is the force acting parallel to the plane, namely

$$(1.27) \qquad \bar{F}_{||} = \delta_T\bar{T}\cos\alpha - \delta_D\bar{D}$$

in terms of scaled thrust $\bar{T}$ and drag $\bar{D}$. Similarly, $\bar{F}_{\perp}$ is the perpendicular lift force

$$(1.28) \qquad \bar{F}_{\perp} = \bar{L} + \delta_T\bar{T}\sin\alpha$$

in terms of the scaled lift $\bar{L}$. For the aircraft used, the parameters had the approximate values $\delta_E = 1.1854$, $\delta_W = .0430$, $\delta_D = .0313$, $\delta_V = .2503$, $\delta_T = .0475$ and $\varepsilon = 3.0516/N$ where N = number of kilometers in the range-to-go.

From this scaling in equations (1.21) - (1.26) we clearly see the separation of x,y from E,h,γ,β for long-range (large N) problems. In addition, note that there is a milder separation of E,h,γ and β. The order of $\frac{d\bar{E}}{d\bar{t}}$ is $\delta_E\delta_D \simeq .04$, the order of $\frac{d\bar{h}}{d\bar{t}}$ is $\delta_V \simeq .25$ and the order of both $\frac{d\gamma}{d\bar{t}}$ and $\frac{d\beta}{d\bar{t}}$ is of order 1. On the basis of this scaling, there appears to be a time scale separation making x,y the slowest variables with E next, h after E and with γ,β together as the fastest. However, two important points are in order. First, in our model aircraft the angle-of-attack α was restricted so that $\alpha \geq 0$. Thus, in equation (1.25) we can only have the dominant term $\bar{F}_\perp$ nonnegative. Practically speaking, this means that the aircraft can accelerate upward much faster than it can accelerate downward. This fact was supported by simulation runs which showed that it takes much more time to decrease γ than to increase it. In effect, if γ must be decreased, then it varies more slowly than β. The second point we wish to make is that time scale separation is also determined by sensitivity of the long term cost-to-go. Thus, it is possible that β should be considered slower than its ranking in equations (1.21) - (1.26) by virtue of its effect on the long range cost-to-go. Unfortunately, we have no method for analyzing the sensitivity of the cost-to-go for such a complex optimization problem as the one at hand. Therefore, we considered both the conventional orderings (x,y), (E,β), h, γ and (x,y), E, β, h, γ as well as the ordering (x,y), E, h, γ, β indicated by the scaling and our discussion given above. For the initial conditions studied we found that the latter ordering yielded faster times than the conventional ones (mainly due to the asymmetric behavior of γ dependent on whether it is increasing or decreasing). Note that if β is considered to be the fastest variable, then the SPT algorithm is considerably simplified. One essentially calculates α and u for vertical plane flight and adjusts σ continuously to head to the target (or the predicted position of the target). The other orderings are considered in Section 5.4 (E faster than β, β faster than h or γ) and Section 5.5 (E,β together faster than h or γ).

In the following sections we will first work through the analysis required for the solution of each of the control subproblems, and then we will summarize the resulting algorithms and note the computation required for their on-board execution. As we noted in Section 4.2, the SPT approximation involves solving a hierarchy of control problems which must be solved in sequence (see Figure

4.2.2). Thus, the total time required to calculate one update of the aircraft controls by means of the SPT method is the sum of the computation times required for each of the algorithms which solve the individual subproblems. In addition to noting the computation time required for the individual subproblem algorithms, we will also note which functions must be stored and which quantities could be obtained directly from aircraft sensors.

The organization of this chapter roughly follows the separation of time scales we have assumed for the aircraft model. Thus, Section 5.2 discusses the x,y reduced order problem, Section 5.3 discusses the E boundary layer problem, and Section 5.4 discusses the β boundary layer problem. Due to the difficulty of computing the β boundary layer solution we discuss the use of a suboptimal solution in Section 5.5. Then in Section 5.6 we discuss the h boundary layer problem and finally, in Section 5.7 we discuss the γ boundary layer problem which operates on the fastest time scale.

In each section we have discussed the computational requirements as well as the analysis required for the solution. In addition, in Sections 5.2 and 5.3 we have discussed the possibility of linearization to solve the faster boundary layer problems corresponding to the subproblems of those respective sections. Thus, in Section 5.2 we consider the possibility of linearizing around the x,y reduced order solution as a nominal value to obtain linear cotrols for the E,β,h,γ boundary layer problem. Likewise, in Section 5.3 we consider linearizing around our suboptimal solution for E to obtain linear controls for the corresponding faster boundary layer β, h and γ.

Figure 5.1.1 shows the complete control logic for long range interception. It involves an iterative calculation of the intercept point and the intercept time, t_f. In addition, the calculation of heading on the cruise arc and controls for the three parts of the trajectory (before cruise, cruise and after cruise) is also performed.

5.2 The Reduced Order Problem in x,y: The Cruise Solution

5.2.1 x,y Reduced Order Control Problem and Feedback Solution

By considering x and y on the same time scale and all the other states in (1.3) - (1.8) on a faster time scale we obtain the reduced order dynamical system

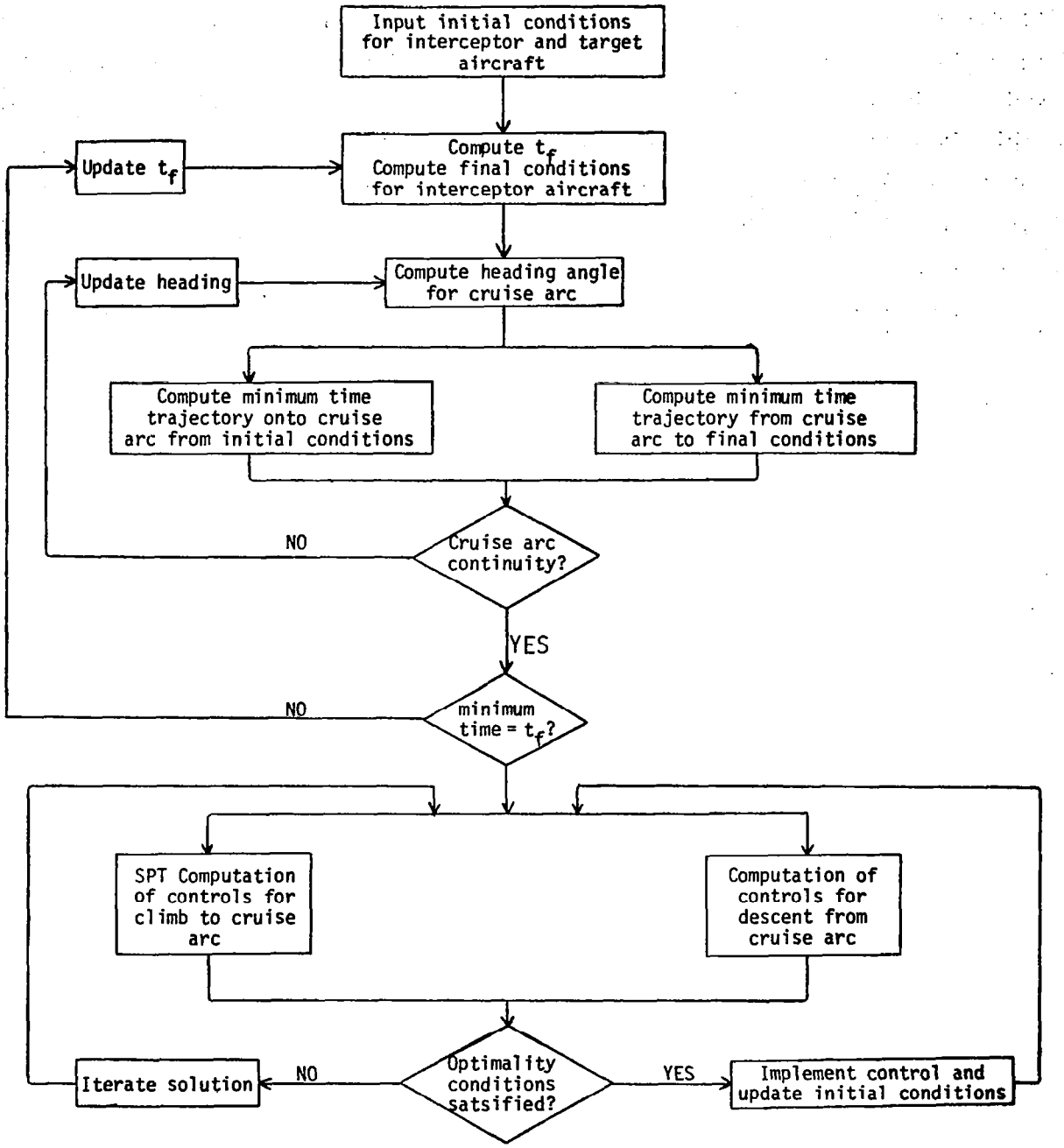


Figure 5.1.1
Overall Control Logic for Long Range Interception

(2.1) $$\frac{dx}{dt} = V\cos\beta\cos\gamma$$

(2.2) $$\frac{dy}{dt} = V\sin\beta\cos\gamma$$

(2.3) $$0 = \frac{V(uT_{max}\cos\alpha - D)}{mg}$$

(2.4) $$0 = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma}$$

(2.5) $$0 = V\sin\gamma$$

(2.6) $$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV}$$

together with the constraints (1.9) - (1.13). The Hamiltonian for this problem is

(2.7) $$H = 1 + \lambda_x V\cos\beta\cos\gamma + \lambda_y V\sin\beta\cos\gamma$$

and since the problem is to minimize time, the Hamiltonian is identically 0 along the optimal trajectory. The pseudocontrols E,h,β,γ and the controls α,σ,u are chosen to maximize the velocity V subject to the equality and inequality constraints and satisfy the x,y terminal conditions. Thus, one easily finds that the optimal pseudocontrols for the reduced order problem are given by:

(2.8) $$E^* = E_c$$

(2.9) $$\beta^* = \tan^{-1}\frac{y_f - y_i}{x_f - x_i}$$

(2.10) $$h^* = h_c$$

(2.11) $$\gamma^* = 0$$

(2.12) $$\alpha^* = \alpha_c$$

(2.13) $$\sigma^* = 0$$

(2.14) $$u^* = u_c$$

where the subscript "c" denotes constant cruise values. Let V_c denote the corresponding maximum cruise velocity determined from E_c and h_c by (1.18). For our particular aircraft, which is taken from Parsons (1972), these constants are given by

$$E_c = 2.9949 \times 10^4 \text{ m} \tag{2.15}$$

$$h_c = 1.2192 \times 10^4 \text{ m} \tag{2.16}$$

$$\alpha_c = 1.509^\circ \tag{2.17}$$

$$u_c = .8928 \tag{2.18}$$

$$V_c = 590.2 \text{ ms}^{-1} \tag{2.19}$$

In order to define the cost criterion for the E boundary layer we need to obtain the adjoints λ_x and λ_y for this reduced order problem. This is easy to do and we find that they are given by the expressions

$$\lambda_x = -\frac{\cos\beta_c}{V_c} \tag{2.20}$$

$$\lambda_y = -\frac{\sin\beta_c}{V_c} \tag{2.21}$$

where we have introduced the notation β_c for the angle given by (2.9) or by

$$\cos\beta_c = \frac{x_f - x_i}{((x_f-x_i)^2 + (y_f-y_i)^2)^{\frac{1}{2}}} \tag{2.22}$$

$$\sin\beta_c = \frac{y_f - y_i}{((x_f-x_i)^2 + (y_f-y_i)^2)^{\frac{1}{2}}} \tag{2.23}$$

Note that (2.22) and (2.23) uniquely define β_c as the heading angle between the initial point (x_i,y_i) and the final point (x_f,y_f), measured from the positive x direction to the line between these points. Note that we can rewrite the

control law for β^* in (2.9) as

(2.24) $\beta^* = \beta_c$

For convenience we will use the simpler expression (2.24) in the future rather than the expression (2.9). However, it should be kept in mind that β_c is a function of the current x and y states as expressed in (2.9) or in (2.22), (2.23)

The cost criterion for the next level of the SPT hierarchy of control subproblems is found by substituting the expressions for the adjoints λ_x and λ_y from (2.20), (2.21) into (2.7) to obtain

(2.25) $\hat{L} = 1 - \frac{V}{V_c} \cos\gamma\cos(\beta-\beta_c)$

5.2.2 Computational Requirement for the Solution

Clearly, the computations required in this reduced order problem are trivial. All the controls and pseudocontrols are constants which are precomputed and stored, except for the pseudocontrol $\beta^* = \beta_c$, which is easily computed using one of the inverse trigonometric functions from (2.9), (2.22) or (2.23). Figure 5.2.1 summarizes the computational requirements at this level.

Storage	Calculation
E_c (constant)	β_c
h_c (constant)	(by means of
α_c (constant)	equation (2.9)
u_c (constant)	or (2.22) or (2.23))

Figure 5.2.1
Computational Requirements of x,y Reduced Order Problem

Note that these are the minimum on-board calculations that must be made and the minimum storage required for the x,y reduced order problem. At faster levels we will assume that E_c, h_c, α_c, u_c have already been stored and that β_c has already been calculated, and we will not count this storage or calculation in

the computational requirement for the faster level solution. In this way, the total on-board computational requirement for the problem will be obtained by adding together the requirements at each level without the risk of counting storage or calculations more than once.

5.2.3 Linearization Around the x,y Reduced Order Solution

In Subsection 5.3.4 we discussed the possibility of linearizing the boundary layer problem around the reduced order solution. In this case we would obtain a linear, quadratic criterion, time invariant, infinite time problem in the perturbations of E, β, h, γ, u, σ and α from their optimal values computed in the x,y reduced order solution. The linear control problem has a simple control law of the form

$$(2.26) \qquad \begin{pmatrix} u \\ \sigma \\ \alpha \end{pmatrix} = \begin{pmatrix} u_c \\ 0 \\ \alpha_c \end{pmatrix} + G \begin{pmatrix} E - E_c \\ \beta - \beta_c \\ h - h_c \\ \gamma \end{pmatrix}$$

where G is a constant 3×4 gain matrix computed off-line.

The equation (2.26) gives us the linear regulator control to maintain the cruise conditions of the aircraft. Thus, on the cruise portion of the trajectory the full control law for the aircraft is particularly simple. Figure 5.2.2 summarizes the on-board computational requirements in this case.

Storage	Calculation
G (a 3×4 array of constants)	u, σ, α (by means of equation (2.26))

Figure 5.2.2
On-Board Computational Requirements During Cruise

5.3 The Initial Boundary Layer Problem in E

5.3.1 E-Boundary Layer Problem and Feedback Solution

By considering E on a faster time scale than x or y and on a slower time scale than β, h or γ, we obtain the following boundary layer dynamical system for E:

$$\frac{dE}{dt} = \frac{V(uT_{max}\cos\alpha - D)}{mg} \tag{3.1}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{3.2}$$

$$0 = V\sin\gamma \tag{3.3}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{3.4}$$

together with the constraints (1.9) - (1.13). The cost criterion for this problem is found from (2.25) to be

$$J = \int_0^\infty [1 - \frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c)]\, d\tau \tag{3.5}$$

The initial condition is given by $E = E_i$ but the final condition is given by $E = E_c$ and <u>not</u> by $E = E_f$. In particular, we see that the resulting SPT control law will fly the plane to the cruise condition but will not yield a control which will give some final energy other than the cruise energy. We have discussed this problem more generally in Subsection 4.4.2. We discuss a possible solution for this particular aircraft problem in Section 5.5 of this chapter.

Since the control problem for this boundary layer has only one state dimension, we may use the technique of Subsection 4.3.2 to solve for the feedback control law by minimizing or maximizing the expression

$$\frac{mg(1 - \frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c))}{V(uT_{max}\cos\alpha - D)} \tag{3.6}$$

with respect to the pseudocontrols β,h,γ and the controls u,σ,α such that the

inequality constraints of (1.9) - (1.13) and the equality constraints of (3.2) - (3.4) are maintained. In addition, if we are minimizing (3.6), then we must restrict ourselves to pseudocontrol and control values such that $\frac{dE}{dt}$ given by (3.1) is positive, and if we are maximizing (3.6), then we must restrict ourselves such that $\frac{dE}{dt}$ is negative.

From the equality constraints it is not hard to deduce that

(3.7) $\gamma = 0$

(3.8) $\sigma = 0$

Thus, the pseudocontrol γ and the control σ are unchanged from their values given by the x,y reduced order solution. Moreover, it is not hard to see that the optimization problem gives

(3.9) $\beta = \beta_c$

so that the pseudocontrol β also has its reduced order value. Substituting (3.7), (3.8) and (3.9) into the original E-boundary layer problem, we obtain the following optimization problem. We must maximize or minimize the function

(3.10) $$\frac{mg(1 - \frac{V}{V_c})}{V(uT_{max}\cos\alpha - D)}$$

subject to the equality constraint

(3.11) $0 = L + uT_{max}\sin\alpha - mg$

together with the inequality constraints (1.9), (1.10), (1.11), (1.12). Using the equality constraint (3.11) to solve for u and substituting this expression back into (3.10) we obtain the expression

(3.12) $$\frac{mg(1 - \frac{V}{V_c})}{V([mg - L]\cot\alpha - D)}$$

which we must maximize or minimize, according to the convention described above

after (3.6). By eliminating u we also obtain a new inequality constraint on V and α corresponding to the original constraint on u in (1.12). The new constraint is

$$(3.13) \qquad 0 \leq \frac{mg - L}{T_{max}\sin\alpha} \leq 1$$

We have now reduced the optimization problem to a problem involving only two variables, V and α. Let us now consider V fixed and examine the dependence of (3.12) on α. Using (1.14) - (1.16) we can write the α dependence explicitly in (3.12) as

$$(3.14) \qquad \frac{mg(1 - \frac{V}{V_c})}{V([mg - L_\alpha\alpha]\cot\alpha - D_0 - L_{\alpha\eta}\alpha^2)}$$

Similarly, the inequality constraint (3.13) becomes

$$(3.15) \qquad 0 \leq \frac{mg - L_\alpha\alpha}{T_{max}\sin\alpha} \leq 1$$

It is not hard to see that for α in the range $0 < \alpha < \pi$ the denominator of (3.14), namely

$$(3.16) \qquad V([mg - L_\alpha\alpha]\cot\alpha - D_0 - L_{\alpha\eta}\alpha^2)$$

is a strictly decreasing function of α. Thus, the expression (3.14) is an increasing function of α in this range. Note that this range easily includes all admissible α since $\alpha_s < \pi$ (in our aircraft example $\alpha_s = 12°$).

Since (3.14) is monotonically increasing in α, in order to maximize (3.14) with respect to α we must choose the largest possible α consistent with the inequality constraints (3.15) and (1.9). Similarly, if we wish to minimize (3.14), we must choose the smallest possible α consistent with the inequality constraints.

Note that the function

$$u(\alpha) = \frac{mg - L_\alpha \alpha}{T_{max} \sin\alpha} \tag{3.17}$$

which appears in the inequality constraint (3.15) is monotonically decreasing in α for admissible α (in fact, for all α such that $0 < \alpha < \frac{\pi}{2}$). The monotonic behavior of the function (3.17) allows us to describe the maximum and minimum allowable values of α fairly easily. There are essentially three different cases to consider.

Case 1. This case corresponds to

$$L_\alpha \alpha_s + T_{max} \sin\alpha_s < mg \tag{3.18}$$

In this case there is no admissible α which satisfies the inequality constraints (3.15) and (1.9). Physically, (3.18) represents the stall constraint for the plane and it defines the region in which the plane cannot maintain vertical equilibrium using both lift and thrust forces.

Case 2. This case corresponds to the two relations

$$L_\alpha \alpha_s < mg \tag{3.19}$$

and

$$T_{max} \sin\alpha_s + L_\alpha \alpha_s \geq mg \tag{3.20}$$

Physically, there is less lift force than the weight of the aircraft, but by using maximum thrust in addition to lift, the plane is able to maintain equilibrium. The maximum admissible α is given by $\alpha = \alpha_s$. The minimum admissible α is given by solving the equation

$$T_{max} \sin\alpha + L_\alpha \alpha = mg \tag{3.21}$$

Case 3. This case corresponds to

$$L_\alpha \alpha_s \geq mg \tag{3.22}$$

Physically, the aircraft has enough lift to maintain vertical equilibrium by

lift force alone. The maximum admissible α is given by

$$(3.23) \qquad \alpha = \frac{mg}{L_\alpha}$$

and the minimum admissible α is given by solving (3.21) as above.

The argument above concerning α allows us to express α as a function of V and E, and hence allows us to eliminate α from the problem to optimize (3.14). Since the aerodynamic quantities L_α, D_0 and T_{max} are given only as data, we must solve the V optimization numerically. We have done this off-line and stored the resulting optimal control laws $V^*(E)$. Note that there are two V^* controls, one corresponding to the maximization and one corresponding to the minimization of (3.14). We have found that for the optimal $V = V^*(E)$, only Case 3 above occurs. Thus, the throttle control u is 0 or 1 depending on whether E is above or below the cruise energy E_c. The angle-of-attack α is chosen to maintain vertical equilibrium, i.e., to satisfy equation (3.23) if $u = 0$ and to satisfy (3.21) if $u = 1$. In Figure 5.3.1 we have summarized the SPT feedback control law for the E-boundary layer.

To obtain the cost criterion for the next level control problem, we need to obtain the optimal adjoint λ_E^* for the E boundary layer problem. This is easily accomplished by solving

$$(3.24) \qquad 0 = 1 - \frac{V^*}{V_c} + \lambda_E^* \frac{V^*}{mg} (T^*\cos\alpha^* - D^*)$$

Note that (3.24) gives λ_E^* as a function of V^*, which in turn is a function of E, namely

$$(3.25) \qquad \lambda_E^* = \frac{mg(\frac{V^*}{V_c} - 1)}{V^*(T^*\cos\alpha^* - D^*)}$$

We may either store λ_E^* explicitly as a function of E or we may compute λ_E^* from $V^*(E)$ using (3.25). However, the latter approach requires us to store the aerodynamic functions T^*, D^*.

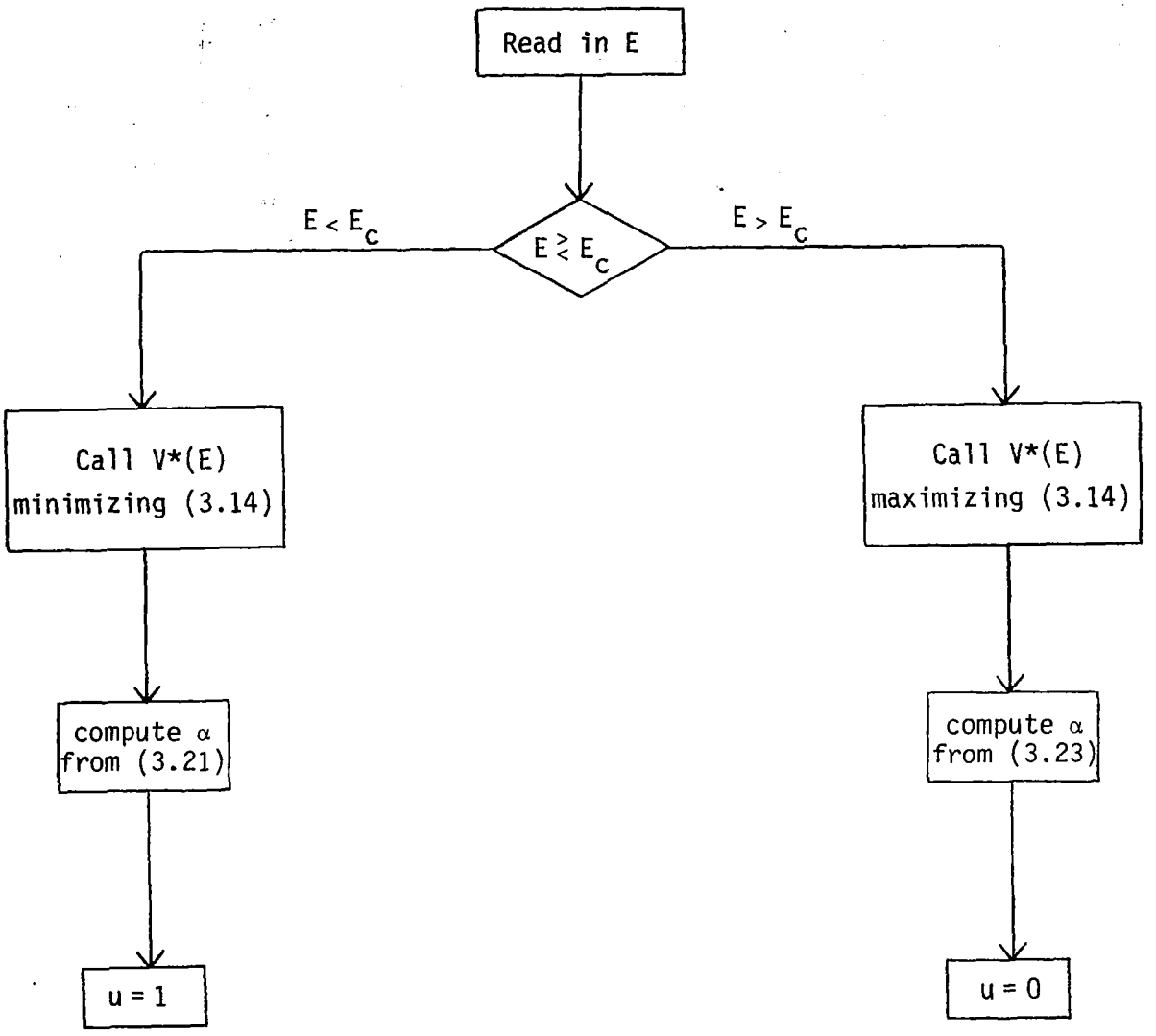


Figure 5.3.1.-E-Boundary Layer Algorithm.

5.3.2 Computational Requirements for the Solution

The important new quantities computed in the E level control problem are the pseudocontrols $V^*(E)$ and $h^*(E)$, the controls $\alpha^*(E)$ and $u^*(E)$, and the adjoint $\lambda_E^*(E)$. We have stored V^* (for both $E < E_c$ and $E > E_c$) as a function of E. From $V^*(E)$ we can easily compute $h^*(E)$ as

$$h^*(E) = E - \frac{V^*(E)^2}{2g} \tag{3.26}$$

The control function $u^*(E)$ is a trivial switch function as can be seen from Figure 5.3.1. For $E > E_c$, we have $u^*(E) = 0$ and for $E < E_c$ we have $u^*(E) = 1$.

The control $\alpha^*(E)$ is obtained from solving (3.21) if $E < E_c$ and from solving (3.23) if $E > E_c$. Equation (3.21) gives α only implicitly so that we must approximate the exact solution. For $L_\alpha \gg mg$, the linear approximation $\sin\alpha \simeq \alpha$ gives an explicit formula,

$$\alpha^*(E) \simeq \frac{mg}{T_{max} + L_\alpha} \tag{3.27}$$

for $\alpha^*(E)$. Using Newton-Raphson's method with (3.27) as an initial guess, we can obtain a better approximation of $\alpha^*(E)$ with only a few iterations.

On-board computation of $\alpha^*(E)$ from either (3.21) or (3.23) will require storing the aerodynamic functions $T_{max}(E,h)$ and $L_\alpha(E,h)$. The function L_α has the form

$$L_\alpha = \tfrac{1}{2} C_{L_\alpha}(M)\rho(h)V^2 S \tag{3.28}$$

where S is a constant, M is Mach number related to velocity by the equation

$$M = \frac{V}{c(h)} \tag{3.29}$$

where c(h) denotes the speed of sound at altitude h. Thus, to store $L_\alpha(E,h)$ it suffices to store functions of only one variable and then compute L_α from these using (3.28).

The function T_{max}, on the other hand, is only given as data and is not given in terms of functions of a single variable. Thus, we must store T_{max} in

an efficient form. At present T_{max} has been approximated using bicubic splines, but the approximation is costly in terms of storage and this method of storing T_{max} should be improved.

Figure 5.3.2 summarizes the computational requirements at the E-boundary layer level.

Storage	Computation
V*(E) (function of one variable) (piecewise linear approximation over five intervals in E)	α^* (from (3.21) or (3.23))
$L_\alpha(E,h)$ (function of two variables)†	h^* (from (3.26))
$T_{max}(E,h)$ (function of two variables)	

Figure 5.3.2.-Computation and Storage Requirements for E-Layer Solution.

5.3.3 Linearization Around the E-Boundary Layer Solution

As in Subsections 4.3.4 and 5.2.3, we can also linearize the boundary layer control problem for β, h and γ around the nominal values for these variables obtained in the E-boundary layer problem solution. As before, we obtain a linear, quadratic cost criterion, time invariant, infinite time problem in the perturbations of the variables β, h, γ, σ, α from their optimal values computed in the E-boundary layer problem. Note that we assume that u retains the same value, either 0 or 1, in the higher level control problems that it has in the E level problem. The linear control problem has a simple control law of the form

$$(3.30) \qquad \begin{pmatrix} \sigma \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha^*(E) \end{pmatrix} + G(E) \begin{pmatrix} \beta-\beta_c \\ h-h^*(E) \\ \gamma \end{pmatrix}$$

where G is a 2×3 gain matrix which depends only on E. In this case the gain matrix G is a function of the energy E and must be computed off-line and stored

†Alternatively, store the functions C_{L_α}, ρ, c of one variable and compute L_α from (3.28).

efficiently for on-board use. Note that storage of G amounts to the storage of six functions of one variable. Figure 5.3.3 summarizes the computational requirements for this linearization.

Storage	Calculation
G	σ,α
(a 2×3 array of functions of one variable)	(by means of equation (3.30))

Figure 5.3.3.-Computation and Storage Requirements for Linearized E-Layer Solution.

5.4 The Initial Boundary Layer Problem in β

5.4.1 β-Boundary Layer Problem and Feedback Solution

By considering β on a faster time scale than x, y or E and on a slower time scale than h or γ, we obtain the following boundary layer dynamical system for β:

$$\frac{d\beta}{dt} = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{4.1}$$

$$0 = V\sin\gamma \tag{4.2}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{4.3}$$

together with the constraints (1.9) - (1.13). The cost criterion for this problem is found to be

$$J = \int_0^\infty \left[1 - \frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c) + \lambda_E^* \frac{V}{mg}(uT_{max}\cos\alpha - D)\right] d\tau \tag{4.4}$$

where λ_E^* is the optimal adjoint computed in (3.25). The initial condition is given by $\beta = \beta_i$ but the final condition is $\beta = \beta_c$. Again, as for the E-boundary layer problem, we see that the SPT control law will fly the plane to the cruise heading but will not yield a control which will give the final heading angle.

The optimal control law is obtained by minimizing or maximizing the expression

$$(4.5)\qquad \frac{[1-\frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c)+\lambda_E^*\frac{V}{mg}(uT_{max}\cos\alpha-D)]mV\cos\gamma}{(L+uT_{max}\sin\alpha)\sin\sigma}$$

with respect to the pseudocontrols h, γ and the controls u, σ, α such that the inequality constraints of (1.9) - (1.13) and the equality constraints (4.2) and (4.3) are satisfied. If $\beta < \beta_c$ then we minimize (4.5) and we must restrict ourselves to pseudocontrol and control values which give a positive value for $\frac{d\beta}{dt}$ in (4.1). Likewise, if $\beta > \beta_c$ then we maximize (4.5) and we must restrict ourselves to pseudocontrols and controls such that $\frac{d\beta}{dt}$ remains negative.

From the equality constraint (4.2) it is not hard to see that

$$(4.6)\qquad \gamma = 0$$

At this stage we will make the additional assumption that u takes on the same value (namely 1 or 0) that it does in the E-boundary layer solution. In general, we assume that if a control variable takes only a maximum or minimum value at a slower level of the SPT hierarchy, then it maintains that value of the control at the faster levels of the hierarchy. Thus, instead of writing uT_{max} we will write T and assume that u is already determined and fixed by the E-boundary layer solution.*

With these assumptions the expression (4.3) becomes

$$(4.7)\qquad 0 = (L+T\sin\alpha)\cos\sigma - mg$$

Using (4.7) to simplify the denominator of (4.5), we obtain

$$(4.8)\qquad \frac{[1-\frac{V}{V_c}\cos(\beta-\beta_c)+\lambda_E^*\frac{V}{mg}(T\cos\alpha-D)]V}{g\tan\sigma}$$

Let us fix V and try to determine the optimum α and σ for a given fixed V. To do this we consider σ a function of α determined by the expression (4.7). Taking

*A comparison with the exact energy state solution using both E and β as states on the same time scale shows that this approximation is valid in all cases except those where level flight cannot be maintained during a turn with zero thrust (see Figure A5.1.9). This special case is handled quite easily in practice by switching to full thrust if level flight cannot be maintained otherwise.

the derivative of (4.8) with respect to α, we obtain

$$(4.9)\qquad \frac{-\lambda_E^* V}{mg}\,(T\sin\alpha + 2\eta L_\alpha \alpha)\,\frac{V}{g}\cot\sigma$$

$$-\left(1-\frac{V}{V_c}\cos(\beta-\beta_c)+\lambda_E^*\frac{V}{mg}[T\cos\alpha - D_0 - \eta L_\alpha \alpha^2]\right)\frac{V}{g}\sin^{-2}\sigma\,\frac{\partial\sigma}{\partial\alpha}$$

The derivative $\frac{\partial\sigma}{\partial\alpha}$ is determined from (4.7) to be

$$(4.10)\qquad \frac{\partial\sigma}{\partial\alpha} = \frac{L_\alpha + T\cos\alpha}{L_\alpha\alpha + T\sin\alpha}\cot\sigma$$

If we assume that α is small and that we can neglect terms higher than linear terms in α, then we can determine the optimum σ by setting the expression (4.9) equal to 0. In this way, we obtain

$$(4.11)\qquad \tan^2\sigma = \frac{1-\frac{V}{V_c}\cos(\beta-\beta_c)+\lambda_E^*\frac{V}{mg}(T-D_0)}{-\lambda_E^* V\,\frac{(T+2\eta L_\alpha)}{(T+L_\alpha)^2}\,mg}$$

In (4.11) one takes the positive or negative root depending on whether $\beta<\beta_c$ or $\beta>\beta_c$. The corresponding value for α is determined from (4.7) using the value for σ from (4.11). Note that we must also test the possibility that σ or α lie on a constraint. One does this simply by setting $\sigma=\pm\sigma_{max}$ and solving (4.7) for the corresponding α or by setting $\alpha=\alpha_s$ and solving (4.7) for σ. We then test these values against the value of σ determined by (4.11) and keep the one which gives the smallest (if $\beta<\beta_c$) or greatest (if $\beta>\beta_c$) value of (4.8).

So far we have only determined the optimum σ and α values in terms of a fixed V. To obtain the optimum V, and thus solve the problem, we must resort to numerical optimization off-line. This optimization is made difficult by the dependence of (4.8) on E and $\beta-\beta_c$. Thus, the computed optimal velocity pseudo-control V* for the β-boundary layer is a function of the following form

$$(4.12)\qquad V^* = V^*(E,\beta-\beta_c)$$

Note that $V^*(E,\beta-\beta_c) = V^*(E,-(\beta-\beta_c))$.

The control law for the β-boundary layer is summarized in Figure 5.4.1.

5.4.2 Computational Requirement for the Solution

The major computational requirement at this level is the storage of $V^*(E,\beta-\beta_c)$. Having stored V^*, we can compute σ^* from (4.11) or from the constraint values and then compute α^* from the expression

$$\alpha^* \simeq \frac{mg\ \cos\sigma^*}{L_\alpha + T_{max}} \tag{4.13}$$

We will also have to compute h^* corresponding to $V^*(E,\beta-\beta_c)$ from (3.26), and we need $\lambda_E^*(E)$ to obtain σ^* from (4.11). If λ_E^* is not stored, then it must be computed in the E-boundary layer by means of equation (3.25). Note that in (3.25) the expression V^* is $V^*(E)$, the optimum V for the E boundary layer problem, and not $V^*(E,\beta-\beta_c)$. The computational requirements at the β-boundary layer level are summarized in Figure 5.4.2.

Storage	Calculation
$V^*(E,\beta-\beta_c)$ (function of two variables)	λ_E^* (from (3.25))
D_0 (function of two variables)†	σ^* (from (4.11))
η (function of one variable)	α^* (from (4.13))

Figure 5.4.2

On-Line Computational Requirements for the β-Boundary Layer

In the next section we take an alternative, simpler approach to the β-boundary layer solution discussed here. We use the work of Parsons (1972) to obtain an exact solution to the reduced order problem in x,y,E,β. Then we approximate this exact solution with a suboptimal solution that essentially divides the E-β plane into regions where zero or maximum thrust is used and all the

†Alternatively, store functions C_{D_0}, ρ, c of one variable and compute D_0.

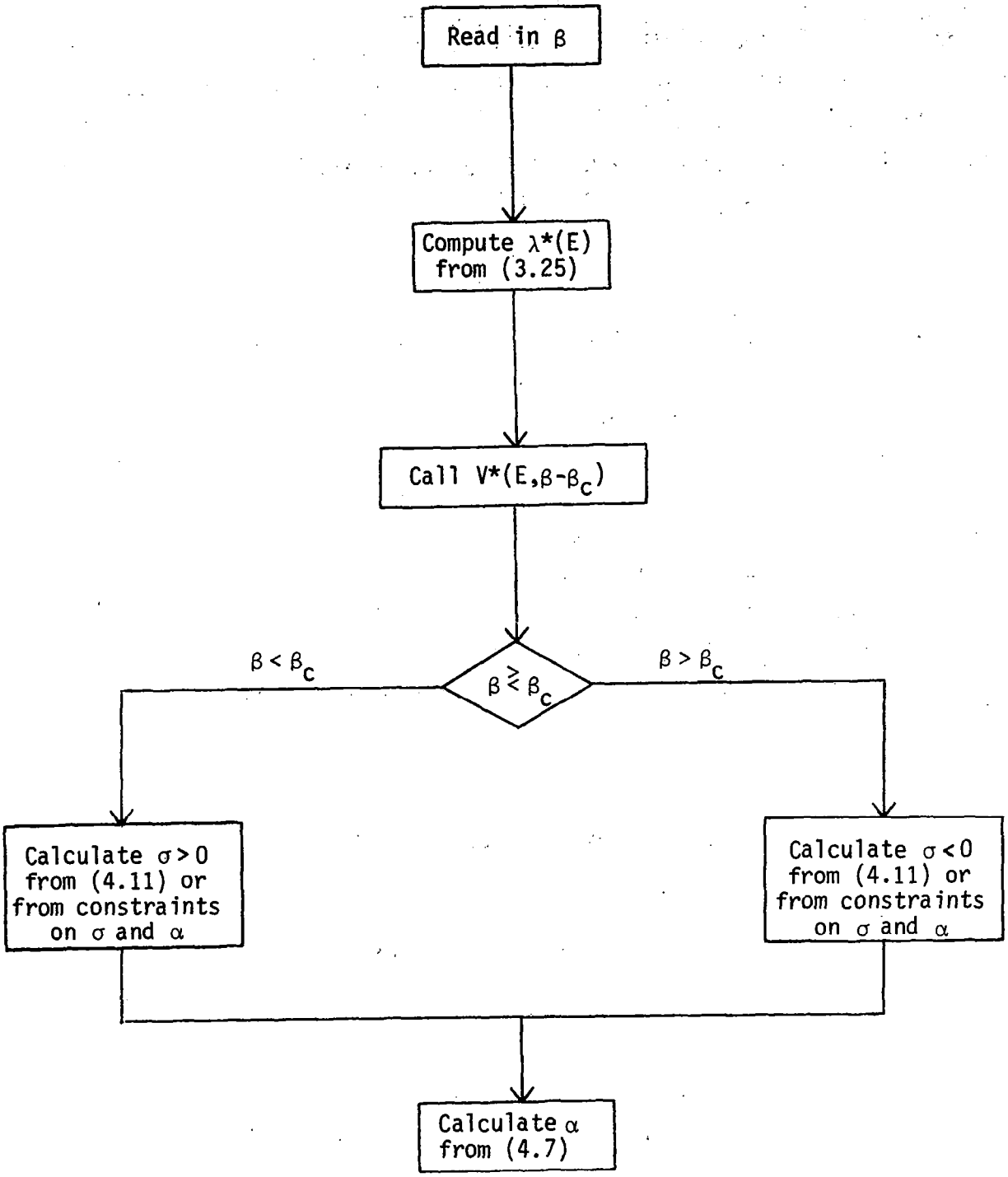


Figure 5.4.1.-β-Boundary Layer Algorithm.

turning takes place either on the maximum turn rate (MTR) locus or on the maximum velocity constraint boundary. This approach involves more storage, but less on-line computation.

Linearization around the β-boundary layer solution for h and γ can be performed in the same way as the linearization around the E-boundary layer (see Subsection 5.3.3).

5.5 E,β Boundary Layer Problem

5.5.1 Exact E,β Solution

Due to the difficulties of computing the β boundary layer feedback control described in Section 5.4, we now suggest an alternative approach. The work of Parsons (1972) provides us with an exact solution to the x,y,E,β problem in which x, y, E and β are treated on the same time scale. Using this exact solution as a basis, we develop a suboptimal feedback law which can be computed on-line and which is not far from optimal.

Assuming that h and γ vary faster than x, y, E, β (which vary on the same time scale), we obtain from (1.3) - (1.8) the dynamic equations

$$\frac{dx}{dt} = V \cos\beta\cos\gamma \tag{5.1}$$

$$\frac{dy}{dt} = V \sin\beta\cos\gamma \tag{5.2}$$

$$\frac{dE}{dt} = \frac{V(uT_{max}\cos\alpha - D)}{mg} \tag{5.3}$$

$$\frac{d\beta}{dt} = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{5.4}$$

with the equality constraints

$$0 = V \sin\gamma \tag{5.5}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{5.6}$$

from (5.5) we have $\gamma = 0$ and from (5.6) we obtain

(5.7) $\quad mg\ \sec\sigma = L + uT_{max}\sin\alpha$

Substituting this expression in (5.4), we have

(5.8) $\quad \frac{d\beta}{dt} = \frac{g\ \tan\sigma}{V}$

Using the fact that α is small and L_α is much greater than T_{max}, we can find an approximate expression for α from (5.7), namely

(5.9) $\quad \alpha = \frac{mg\ \sec\sigma}{L_\alpha}$

Substituting this into (5.3) and approximating $uT_{max}\cos\alpha$ by uT_{max}, we obtain

(5.10) $\quad \frac{dE}{dt} = \frac{V(uT_{max} - D_0 - D_L\sec^2\sigma)}{mg}$

where D_L is given by

(5.11) $\quad D_L = \frac{\eta(mg)^2}{L_\alpha}$

Thus, the system (5.1) - (5.4) becomes

(5.12) $\quad \frac{dx}{dt} = V\cos\beta$

(5.13) $\quad \frac{dy}{dt} = V\sin\beta$

(5.14) $\quad \frac{dE}{dt} = \frac{V(uT_{max} - D_0 - D_L\sec^2\sigma)}{mg}$

(5.15) $\quad \frac{d\beta}{dt} = \frac{g\ \tan\sigma}{V}$

There are inequality constraints on σ, directly through (1.11) and indirectly through (1.9) and (5.9).

This optimal control problem (minimum time) must be solved numerically in terms of a TPBVP derived from the first order necessary conditions. In Appendix 5.1 we set up the TPBVP and discuss the optimization procedure. In the next

section we describe the suboptimal approximation we have chosen for this solution.

5.5.2 Suboptimal Real-Time Approximation

This subsection deals with the problem of computing in real time a suboptimal minimum-time trajectory to intercept a moving object, by approximating the work of Appendix 5.1. These sections have shown that optimization on-line is too time-consuming even for the energy-state approximation and therefore not feasible in real time. A sacrifice in accuracy is called for so that a near-optimal solution may be generated on-line merely by looking up stored trajectories and fitting together portions of them to match the particular boundary conditions imposed upon the aircraft states. The program RLTIME has been written to produce such a near-optimal trajectory. Boundary layers can then be added to this solution in order to smooth out the sudden changes in the states of the aircraft introduced by the energy-state approximation.

Stored Trajectories. Four trajectories are stored:

(1) Max-turn locus with $9022 \leq E \leq 29870$ m. This locus has been defined in Appendix 5.1--basically it requires $\sigma = \sigma_m = \sigma_s$, $\alpha = 12°$ and thrust is maintained either at $T = T_{max}$ or at $T = 0$. The upper limit on E was chosen arbitrarily while the lower limit is approximately the Lufbery Circle Point value (Parsons (1972)). This is the classical aerial combat situation and is a steady-state turning rate condition. If the airplane were to fly along this locus its energy would decrease while following a spiral-shaped trajectory in three-dimension space. The following equations were integrated to obtain the locus with $T = T_{max}$:

$$\frac{d\beta}{dE} = \frac{\tan\sigma_m g W}{V_{MT}^2 (T_{max} - D_0 - D_L \sec^2\sigma_m)} \tag{5.16}$$

$$\frac{dt}{dE} = \frac{V}{V_{MT} (T_{max} - D_0 - D_L \sec^2\sigma_m)} \tag{5.17}$$

$$\frac{dx}{dE} = \frac{W \cos\beta}{(T_{max} - D_0 - D_L \sec^2\sigma_m)} \tag{5.18}$$

$$(5.19) \qquad \frac{dy}{dE} = \frac{W \sin\beta}{(T_{max} - D_0 - D_L \sec^2\sigma_m)}$$

Equations (5.16) - (5.19) have been derived from the equations of motion of the airplane model (5.12) - (5.15). Note that for $9022 \leq E \leq 29870$ m, $T_{max} <$ DRAG and so for $\sigma = +\sigma_m$, β increases as energy decreases. V_{MT} is the velocity necessary to remain on the locus and is given by the equation:

$$(5.20) \qquad V_{MT} = \left(\frac{W}{\frac{1}{2}\rho S C_{L_\alpha} \alpha_s \cos\sigma_m} \right)^{\frac{1}{2}}$$

which merely states that LIFT = W secσ at maximum bank angle and angle-of-attack, i.e., setting $\sigma_s = \sigma_m$.

In order that the x and y values obtained by integration of equations (5.18) and (5.19) may be translated into the inertial coordinate system, consider the following:

$\frac{d\beta}{dE} = f(E)$ from (5.16), hence

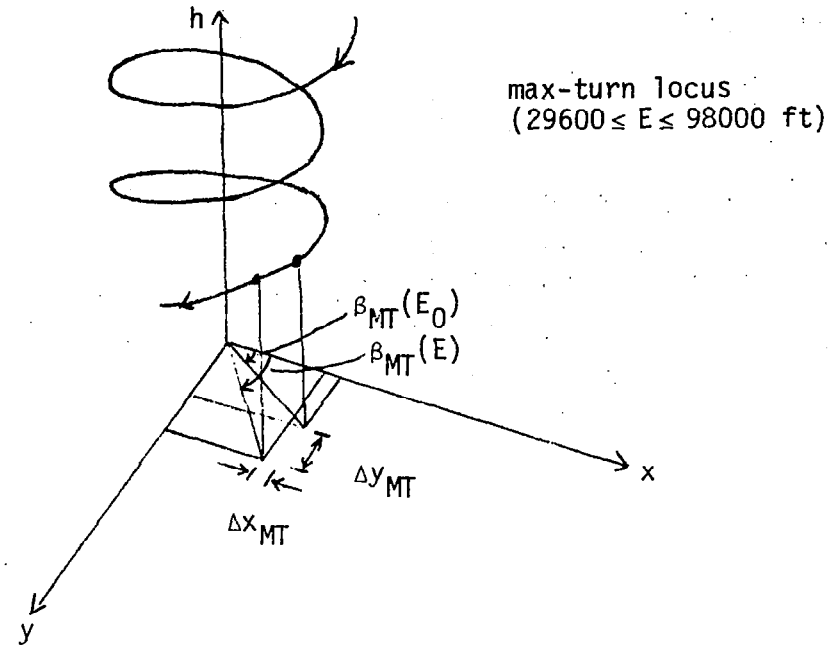
$$(5.21) \qquad \beta(E_1) - \beta(E_0) = \beta_{MT}(E_1) - \beta_{MT}(E_0) = \int_{E_0}^{E_1} f(E)\, dE$$

where the subscript MT refers to the max-turn locus coordinate system and the unsubscripted variables refer to the inertial coordinate system. (See Figure 5.5.1.) Equation (5.18) may now be rewritten as:

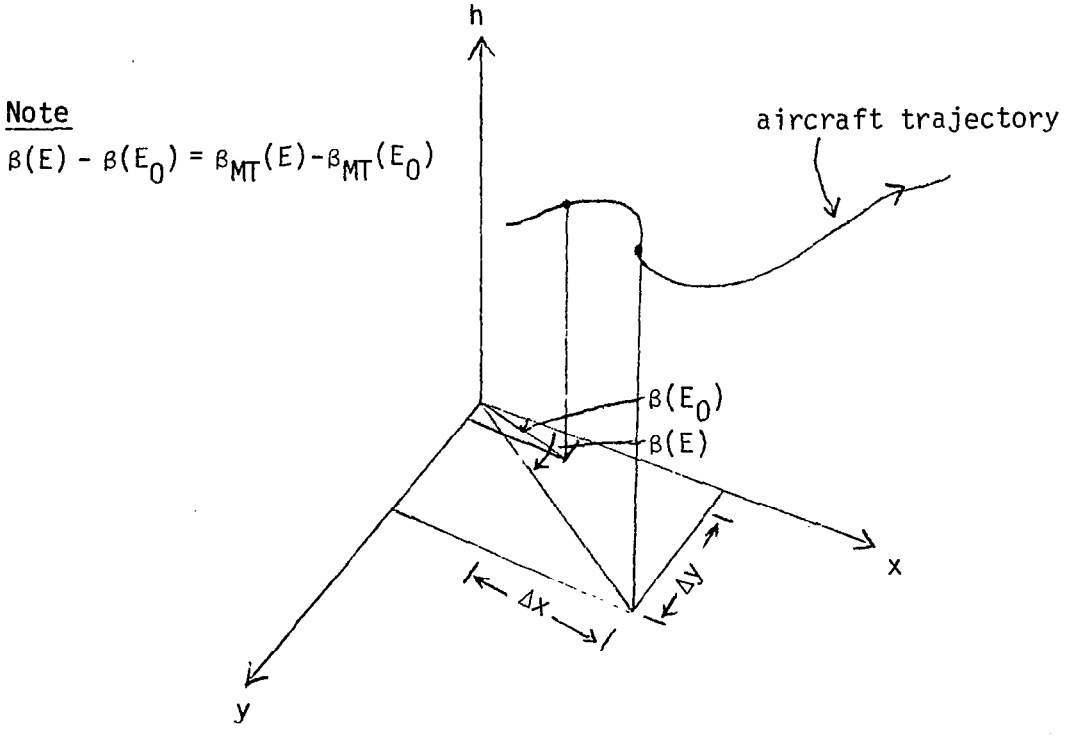
$$(5.22) \qquad \frac{dx}{dE} = - \frac{W \cos[\beta_{MT}(E) + (\beta(E_0) - \beta_{MT}(E_0))]}{(T - D)}$$

$$(5.23) \qquad = \cos(\beta(E_0) - \beta_{MT}(E_0)) \left[\frac{-W \cos\beta_{MT}(E)}{T - D} \right] - \sin(\beta(E_0) - \beta_{MT}(E_0)) \left[\frac{-W \sin\beta_{MT}(E)}{T - D} \right]$$

Hence,



max-turn locus coordinate system



aircraft coordinate system

Figure 5.5.1.-Max-Turn Locus Coordinate System/Aircraft Coordinate System.

$$(5.24)\quad x(E) - x(E_0) = \cos(\beta(E_0) - \beta_{MT}(E_0))[x_{MT}(E) - x_{MT}(E_0)] + \sin(\beta(E_0) - \beta_{MT}(E_0))[y_{MT}(E) - y_{MT}(E_0)]$$

and similarly:

$$(5.25)\quad y(E) - y(E_0) = \cos(\beta(E_0) - \beta_{MT}(E_0))[y_{MT}(E) - y_{MT}(E_0)] + \sin(\beta(E_0) - \beta_{MT}(E_0))[x_{MT}(E) - x_{MT}(E_0)]$$

(2) Max-turn locus with $8992 \leq E \leq 1524$ m. Here the lower limit is arbitrary while the upper limit is again approximately the Lufbery Circle Point value. To obtain the turning in the same sense as in (1) with σ + ve, it is necessary to move up the locus with increasing energy since now $T_{max} > DRAG$. Equations (5.16) - (5.19) were integrated and again equations (5.24) and (5.25) give the horizontal plane distances in the inertial frame.

(3) The $\Delta\beta = 0$, minimum-time energy path to the cruise arc. This is a trajectory belonging to the family of trajectories presented as initial turns to the cruise arc in Appendix 5.1. This particular one allows no turning and is really just the vertical plane solution of the problem of reaching the cruise arc in minimum time.

(4) The $\Delta\beta = 0$ chatter path off the cruise arc. Appendix 5.1 presents the theory supporting this trajectory, which requires zero thrust and is a maximum deceleration arc.

Real-Time Approximations. As stated earlier, a real-time solution requires minimum computation on-line and should involve mainly looking up of stored trajectories and fitting together portions of them so as to satisfy boundary conditions. One possible approximation for real-time determination of trajectories is to turn only on the max-turn locus and at other times to follow either the delta-beta = 0 path to the cruise arc or the chatter path off the cruise arc. Hence, for initial turns to the cruise arc, the approximate solution requires a zoom to the max-turn locus from the initial conditions followed by turning on the locus and then a zoom to the delta-beta = 0 path to the cruise arc.

As presented in Appendix 5.1, Figure A5.1.9 shows the optimal trajectories in the beta-energy plane for final turns from the cruise arc. It is possible

to make approximations in varying degrees to produce near-optimal trajectories to replace the optimal ones, in real time. Using the same approximation as for the initial turns, all the final turning is also on the locus. Thus the real-time approximation requires bank angle chatter on getting off the cruise arc followed by a zoom to the max-turn locus. After turning on the locus a zoom at constant energy to the final conditions completes the trajectory. This approximation is indicated by the dotted lines in Figure 5.5.2 where the thrust-switch locus is also ignored: $T = T_{max}$ all the way once chatter is over.

Numerical results have been obtained for the approximation described above. Probably, a better approximation would be as in Figure 5.5.3 where a piecewise linear fit is made for the thrust-switch locus and the zoom-climb locus and the trajectory allows turning before reaching the max-turn locus by about 60°. The rate of turning may be approximated as constant by the almost straight line trajectories, or alternatively a crude $\frac{dE}{d\beta}$ integration may be performed on-line. Switching the thrust would require storage of the max-turn locus with thrust = 0.

Formulation of the Problem. We assume for simplicity that the target is moving in a straight line at constant velocity. For interception of the target, we require pursuer trajectory time (t_1) = target time to interception point (t_f). In addition $t_1 = t_f$ must be the minimum possible time. Hence an iterative loop is necessary to match up t_1 and t_f for minimum-time interception. With a value for t_f, the position of the target in three-dimensional space may be obtained and the problem may be restated as: given initial and final energies, altitude, velocity, and horizontal plane positions, a minimum-time trajectory is required so that the above boundary conditions are met.

Figure 5.5.4 shows how the min-time path is split up among the stored trajectories and the cruise arc. The heading angle, β, is measured clockwise from the x-axis. Two constraints have to be observed. Restricting ourselves for simplicity to the case of clockwise turns, (1) the relative angle between the initial and final points (β_s) is assumed greater than β_0, the initial heading angle; (2) the total turning angle is assumed less than 180°

In general terms the algorithm is:

(1) Estimate t_f, the interception time
(2) compute final conditions for interceptor aircraft
(3) from initial altitude, execute a velocity (or altitude) zoom (constant

——•——•—— Suboptimal approximation (all turning on max-turn locus)

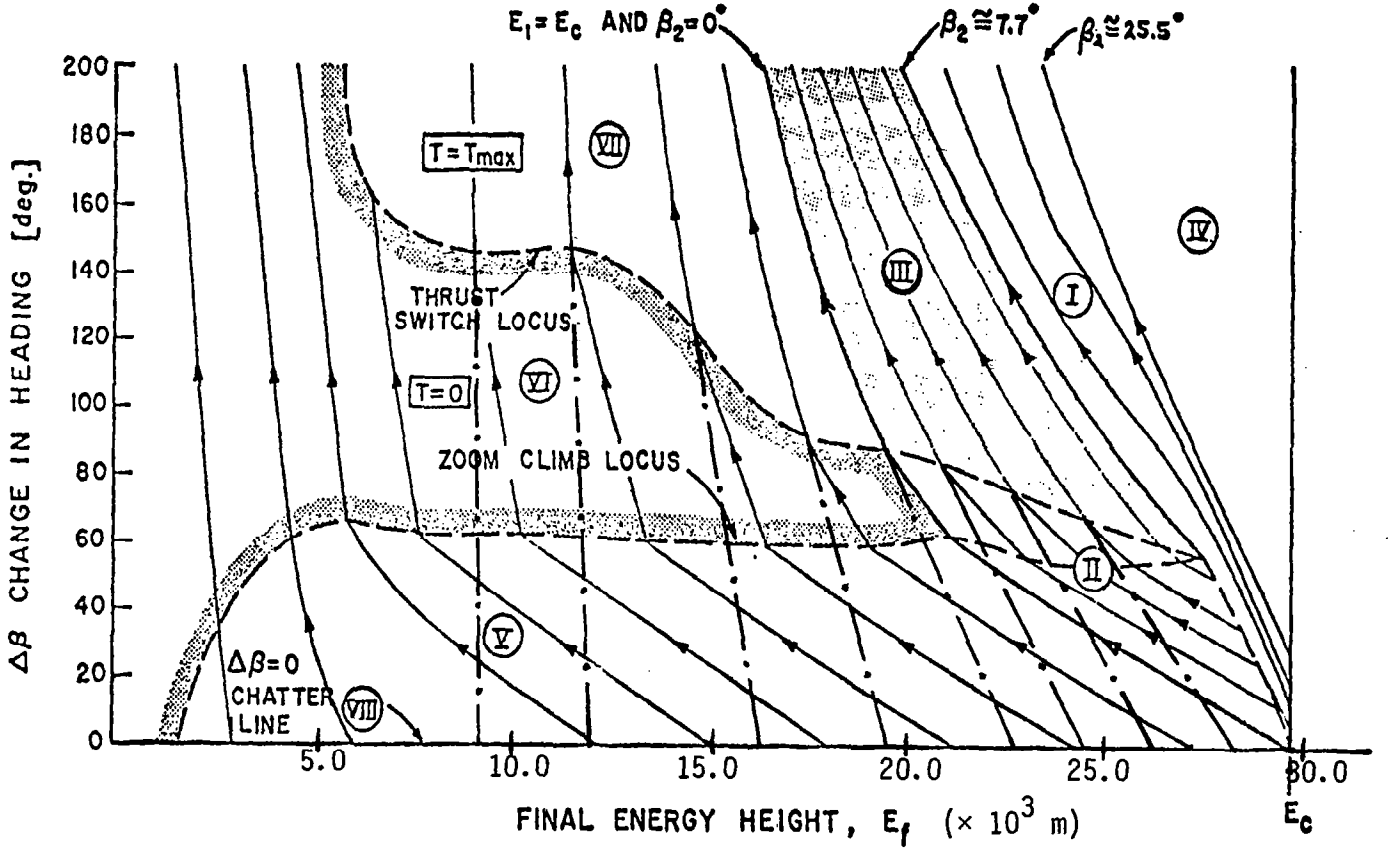


Figure 5.5.2 Change in Heading for Variable-Altitude Final Turns from Cruise Arc

—·—·— Suboptimal approximation (linear fit for thrust switch and zoom climb loci)

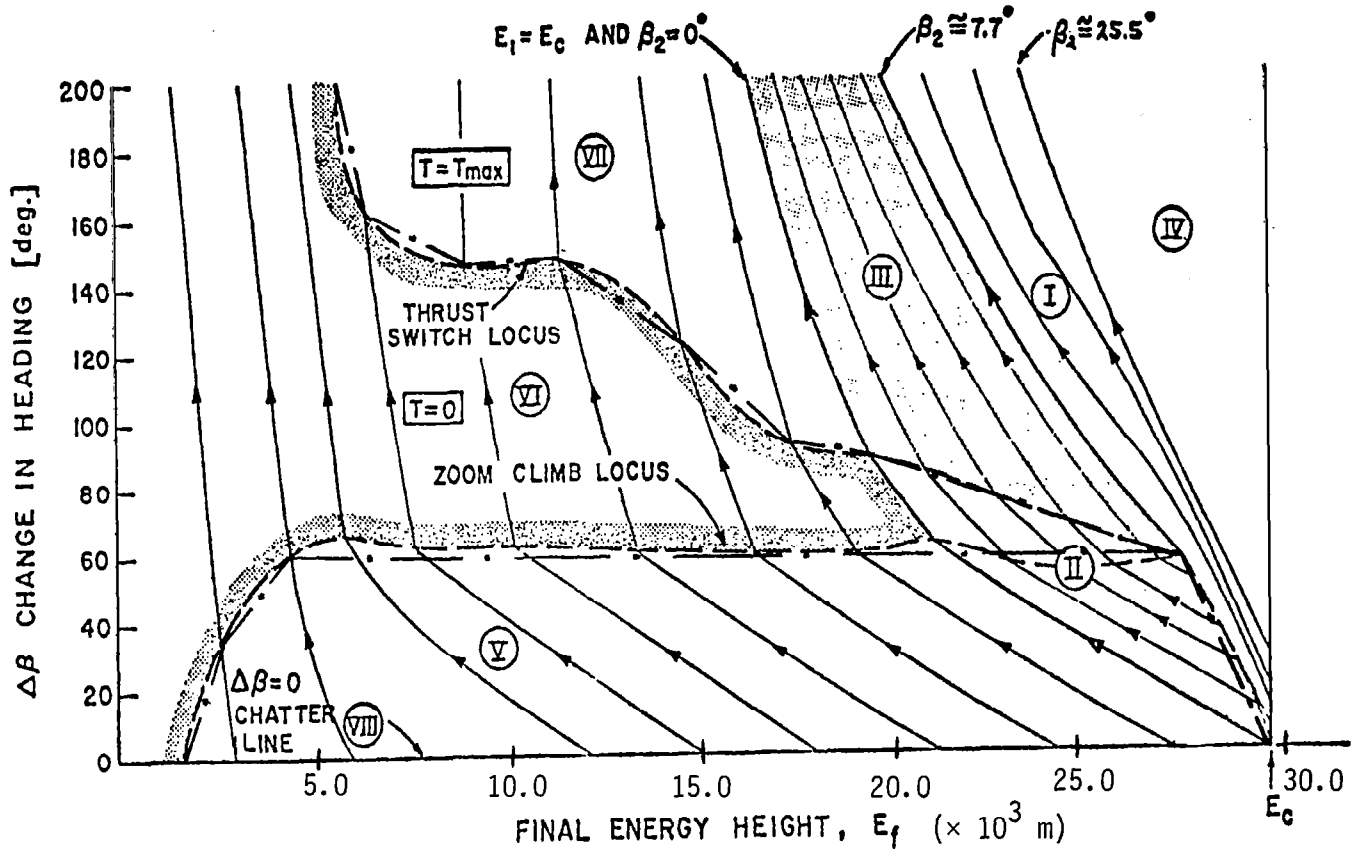


Figure 5.5.3 Change in Heading for Variable-Altitude Final Turns from Cruise Arc

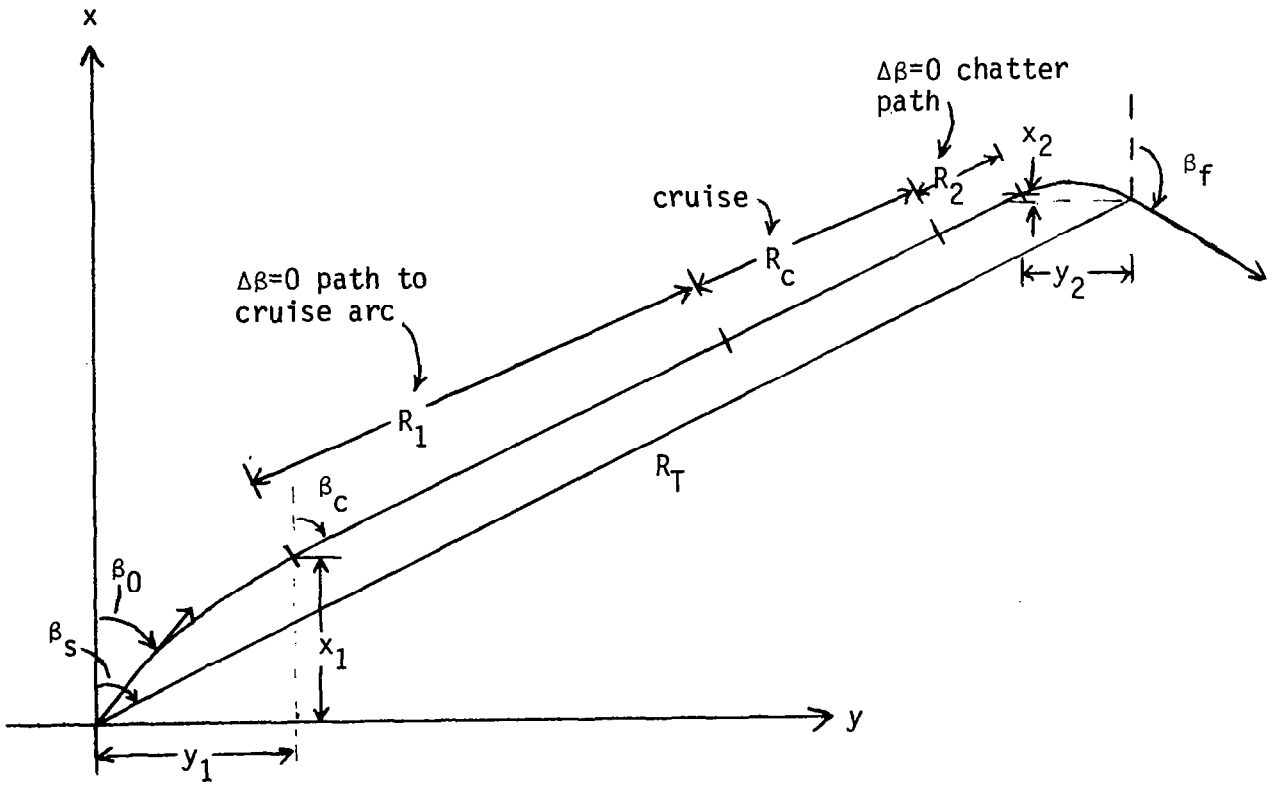


Figure 5.5.4.-Geometry for Determining Initial Heading Angle and Interception Time t_f.

energy) to the max-turn locus

(4) determine cruise heading β_c (see equation (5.28) below)

(5) turn along the max-turn locus until heading = β_c

(6) zoom to the $\Delta\beta = 0$, min-time E-path to cruise arc

(7) follow $\Delta\beta = 0$, min-time E-path to cruise arc

(8) stay on the cruise arc for distance R_c

(9) follow $\Delta\beta = 0$ chatter path off the cruise arc for distance R_2

(10) zoom dive or climb to max-turn locus

(11) follow max-turn locus until heading angle = β_f

(12) zoom dive or climb to required final altitude and velocity

(13) compute trajectory time and update t_f using Newton-type step

(14) go to (2) if accuracy on t_f not met.

Figure 5.5.8 introduces relationships between the various distances involved:

$$(5.26) \qquad y_1 + y_2 + (R_1 + R_2 + R_c)\sin\beta_c = y_f - y_0$$

$$(5.27) \qquad x_1 + x_2 + (R_1 + R_2 + R_c)\cos\beta_c = x_f - x_0$$

which give the cruise heading as:

$$(5.28) \qquad \tan\beta_c = \frac{y_f - y_0 - y_1 - y_2}{x_f - x_0 - x_1 - x_2}$$

and since x_1, y_1, x_2, y_2 are functions of β_c, equation (5.28) has to be solved iteratively, for example by using a Newton method. R_c is then obtained from equation (5.26) or (5.27).

5.5.3 Computational Requirements

The two important quantities being calculated in this algorithm are β_c and t_f, both by iterative methods involving just a few iterations each. We consider here computational requirements for calculation of each of the above two quantities which are sufficient for the second (more accurate) real-time approximation mentioned in the last section. Most of the trajectories to be stored may be approximated by piecewise linear fits to keep storage requirements down. Figure 5.5.5 summarizes the computational requirements.

In the example case presented and considered in Chapter 6, $\beta_c = 38.3°$ while

Storage	Computation
$T = 0$ } $\Delta x(\beta,E)$ $T = T_{max}$ } $\Delta y(\beta,E)$ on max-turn locus	β_c using equation (5.28) in iterative loop
thrust switch locus (β,E)	
β_c	t_f iteratively using t_1 = sum of times on separate portions of trajectory; t_c time on cruise = $\frac{R_c}{590.2}$
$T = 0$ $T = T_{max}$ } $\Delta t(E)$, $h(E)$ on max-turn locus	
thrust switch locus (β,E) zoom climb locus (β,E) β-E lines on max-vel constraint (see Figure 5.5.2)	
$h(E)$, $\Delta t(E)$, $\Delta x(E)$ on $\Delta\beta = 0$ path to cruise	
$h(E)$, $\Delta t(E)$, $\Delta x(E)$ on $\Delta\beta = 0$ chatter line (max-vel constraint)	

Figure 5.5.5.-Storage and Computation Requirements for E,β Layer.

the relative angle between the initial and final points was 38.65°. Hence, the β_c calculation may be omitted to save computation time with little loss in accuracy.

5.6 Initial Boundary Layer Problem in h

5.6.1 h-Boundary Layer Problem and Feedback Solution

As we discussed previously in Sections 5.4 and 5.5, we have separated the vertical plane part of the trajectory from the horizontal turn part. Thus, in considering the h-boundary layer we need only look at the vertical plane case with $\sigma = 0$ and $\beta = \beta_c$. Moreover, we assume that any controls which take their value on a constraint in slower levels retain the constraint value at faster levels.* Consequently, we assume that T is determined ($u = 0$ if $E > E_c$ and $u = 1$ if $E < E_c$). With these assumptions the dynamical system for the h boundary layer becomes

$$(6.1) \qquad \frac{dh}{d\tau} = V \sin\gamma$$

with the equality constraint

$$(6.2) \qquad 0 = (L + T \sin\alpha) - mg \cos\gamma$$

and the cost criterion

$$(6.3) \qquad J = \int_0^\infty \left[1 - \frac{V}{V_c} \cos\gamma + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D)\right] d\tau$$

where λ_E^* is the optimal adjoint computed in (3.25). The initial condition is $h = h_i$ and the final condition is $h = h^*(E)$, where h^* is determined from (3.26).

The optimal control law is obtained by minimizing or maximizing the expression

*More general solutions without these assumptions are given in the appendices. In particular, it is shown that, for turning maneuvers with $\sigma \neq 0$, the results of Sections 5.6 and 5.7 are extended quite easily.

$$(6.4) \qquad \frac{[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2)]}{V \sin\gamma}$$

with respect to the pseudocontrol γ and the control α such that the equality constraint (6.2) is satisfied. If $h < h^*(E)$, then we minimize (6.4) and we must restrict ourselves to $\gamma > 0$. If $h > h^*(E)$, then we maximize (6.4) and we must restrict ourselves to $\gamma < 0$. Note that the expression (6.4) is an odd function of γ and that the expression (6.2) is an even function of γ. It follows that the minimum solution $\gamma > 0$ is the negative of the maximum solution $\gamma < 0$ and that α is the same in both cases. Thus, we restrict our attention to the case when $\gamma > 0$.

To determine the solution $\gamma > 0$ that minimizes the expression (6.4), define the Lagrangian function $\mathscr{L}$ as

$$(6.5) \qquad \mathscr{L} = \frac{[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2)]}{V \sin\gamma} + \frac{\lambda[L_\alpha \alpha + T\sin\alpha - mg\cos\gamma]}{V \sin\gamma}$$

where λ is an undetermined Lagrange multiplier. Setting $\frac{\partial \mathscr{L}}{\partial \gamma} = \frac{\partial \mathscr{L}}{\partial \alpha} = 0$, we obtain the relations

$$(6.6) \qquad 0 = \left(\frac{V}{V_c} + \lambda mg\right) - (1 + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \alpha^2 \eta L_\alpha) + \lambda(L_\alpha \alpha + T\sin\alpha))\cos\gamma$$

$$(6.7) \qquad 0 = -\lambda_E^* \frac{V}{mg}(T\sin\alpha + 2\eta L_\alpha \alpha) + \lambda(L_\alpha + T\cos\alpha)$$

Assuming that α is small enough to neglect in (6.6), (6.7),* we find that λmg in (6.6) is negligible and we obtain the following expression for $\cos\gamma$

$$(6.8) \qquad \frac{\frac{V}{V_c}}{1 + \lambda_E^* \frac{V}{mg}(T - D_0)} = \cos\gamma$$

We solve (6.8) for $\cos\gamma$ and then obtain α from (6.2). Note that for values of h

*This approximation is quite good for $T = T_{max}$, but needs refinement for $T = 0$ case.

close to h*, the expression (6.8) may exceed 1 slightly (by .02) due to the absence of the nonzero α in the denominator. In this case we can either set $\gamma^* = 0$, since it will be very close to 0, or we can use a method of successive approximation to obtain a better approximation--for example, solve (6.2) with $\gamma = 0$ and use this approximation of α in (6.6), (6.7) to obtain a better approximation of γ. In Appendix 5.2 we have carried out a more careful analysis of the solution of equations (6.2), (6.6), (6.7) using perturbation analysis. Figure 5.6.1 summarizes the control law computations for the h-boundary layer.

The adjoint λ_h^* for the h-boundary layer is obtained from solving

$$(6.9) \qquad 0 = 1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2) + \lambda_h^* V \sin\gamma$$

for λ_h^*, where γ and α take their optimum values for the h-boundary layer problem.

5.6.2 Computational Requirements

The computational requirements at this stage are rather minimal. All storage requirements have been accounted for on slower levels. Likewise, the adjoint λ_E^* has already been calculated. Hence, the only computational requirement is to calculate γ from (6.8) and α from (6.2). Figure 5.6.2 summarizes these minor requirements.

5.7 Initial Boundary Layer Problem in γ

5.7.1 γ-Boundary Layer Problem and Feedback Solution

We make the same assumptions for this problem that we did for the h-boundary layer. In particular, we assume that T is fixed, and $\sigma = 0$ and $\beta = \beta_c$. The dynamic system for the γ-boundary layer becomes

$$(7.1) \qquad \frac{d\gamma}{d\tau} = L_\alpha \alpha + T\sin\alpha - mg\cos\gamma$$

with the cost criterion

$$(7.2) \qquad J = \int_0^\infty \left[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D) + \lambda_h^* V \sin\gamma\right] d\tau$$

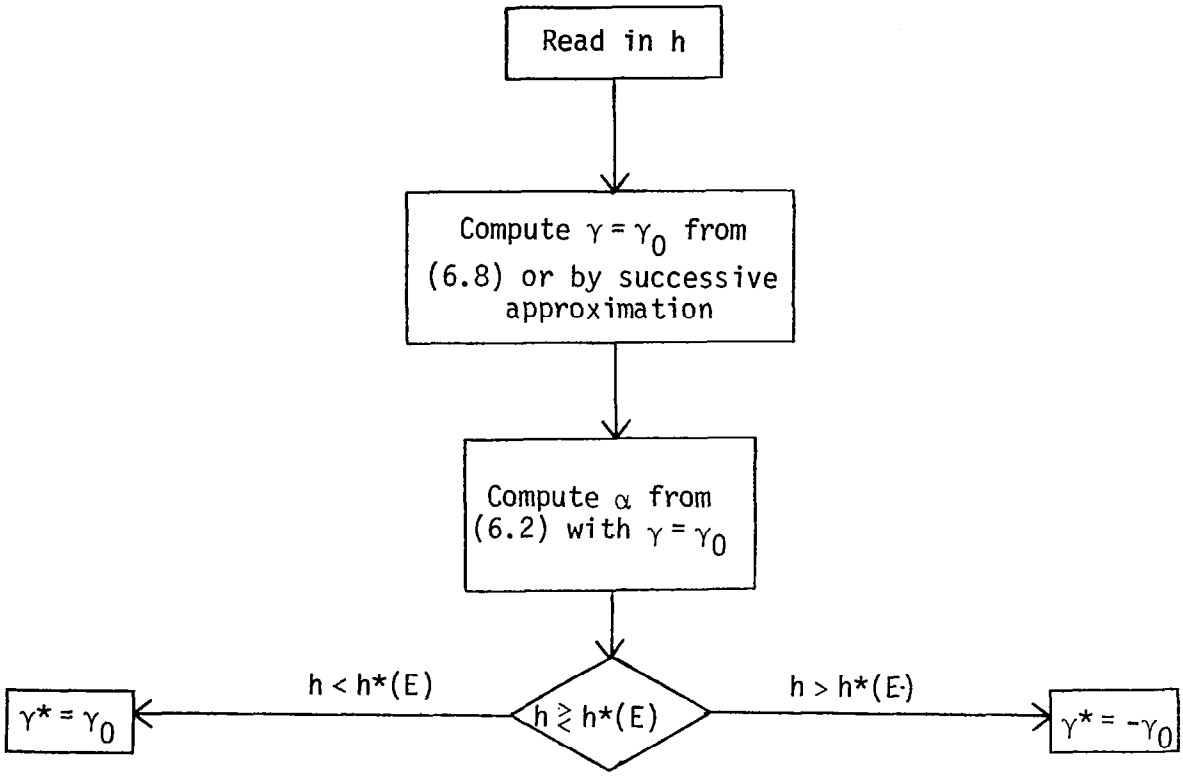


Figure 5.6.1.-h-Layer SPT Algorithm.

Storage	Calculation
None (accounted for at slower levels)	γ^* (from (6.8))
	α^* (from (6.2))

Figure 5.6.2.-Storage and Computation Requirements for h-Layer.

The optimal control law is obtained by minimizing or maximizing the expression

$$(7.3) \qquad \frac{1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2) + \lambda_h^* V \sin\gamma}{L_\alpha \alpha + T\sin\alpha - mg\cos\gamma}$$

with respect to the control α. If $\gamma < \gamma^*(E,h)$, then we minimize (7.3) and we must restrict ourselves to α such that $L_\alpha \alpha + T\sin\alpha - mg\cos\gamma > 0$. If $\gamma > \gamma^*(E,h)$, then we must maximize (7.3) and we must restrict ourselves to α such that $L_\alpha \alpha + T\sin\alpha - mg\cos\gamma < 0$.

Taking the derivative of (7.3) gives us

$$(7.4) \qquad \{-(L_\alpha \alpha + T\sin\alpha - mg\cos\gamma)(\lambda_E^* \frac{V}{mg}[T\sin\alpha + 2\eta L_\alpha \alpha])$$

$$-(L_\alpha + T\cos\alpha)(1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D) + \lambda_h^* V\sin\gamma)\}(L_\alpha \alpha + T\sin\alpha - mg\cos\gamma)^{-2}$$

When either $E > E_c$ and $\gamma < \gamma^*$ or when $E < E_c$ and $\gamma > \gamma^*$, the expression (7.4) is negative. Hence, in these cases the expression (7.3) is decreasing in α. When $E < E_c$ and $\gamma > \gamma^*$, we want to maximize (7.3) so we choose $\alpha = 0$. When $E > E_c$ and $\gamma < \gamma^*$, we want to minimize (7.3) so we choose $\alpha = \alpha_s$.

To solve for α in the other cases, we set (7.4) equal to 0 and we neglect terms in α higher than linear.* Define α_0 by

$$(7.5) \qquad \alpha_0 = \frac{mg\cos\gamma}{L_\alpha + T}$$

Then setting (7.4) equal to 0 gives the following quadratic equation,

$$(7.6) \qquad \alpha^2 - \alpha_0 \alpha + k = 0$$

where k is given by

*In Appendix 5.3 we have considered a higher order approximation.

$$(7.7) \qquad k = \frac{1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T - D_0) + \lambda_h^* V \sin\gamma}{\lambda_E^* \frac{V}{mg}(T + 2\eta L_\alpha)}$$

Note that $k < 0$ for $E < E_c$ and $k > 0$ for $E > E_c$. The solution α is given by

$$(7.8) \qquad \alpha = \frac{\alpha_0}{2} + \tfrac{1}{2}\sqrt{\alpha_0^2 - 4k}$$

when $E < E_c$, and

$$(7.9) \qquad \alpha = \frac{\alpha_0}{2} - \tfrac{1}{2}\sqrt{\alpha_0^2 - 4k}$$

when $E > E_c$. Figure 5.7.1 summarizes the feedback law in this case.

5.7.2 Computational Requirement

As for the h-boundary layer, the computational requirements at this stage are rather minimal. All storage requirements have been accounted for on slower levels. Likewise, the adjoint λ_E^* has already been calculated. The adjoint λ_h^* must be calculated from (6.9) at this level. Finally, α is calculated from (7.8) or (7.9). The computational requirements are summarized in Figure 5.7.2.

Storage	Calculation
None (all accounted for on slower levels)	λ_h^* (from (6.9))
	α_0 (from (7.5))
	k (from (7.7))
	α (from (7.8) or (7.9))

Figure 5.7.2.-Storage and Computation Requirements for γ-Layer.

A pictorial representation of the control problem is given in Figure 5.7.3.

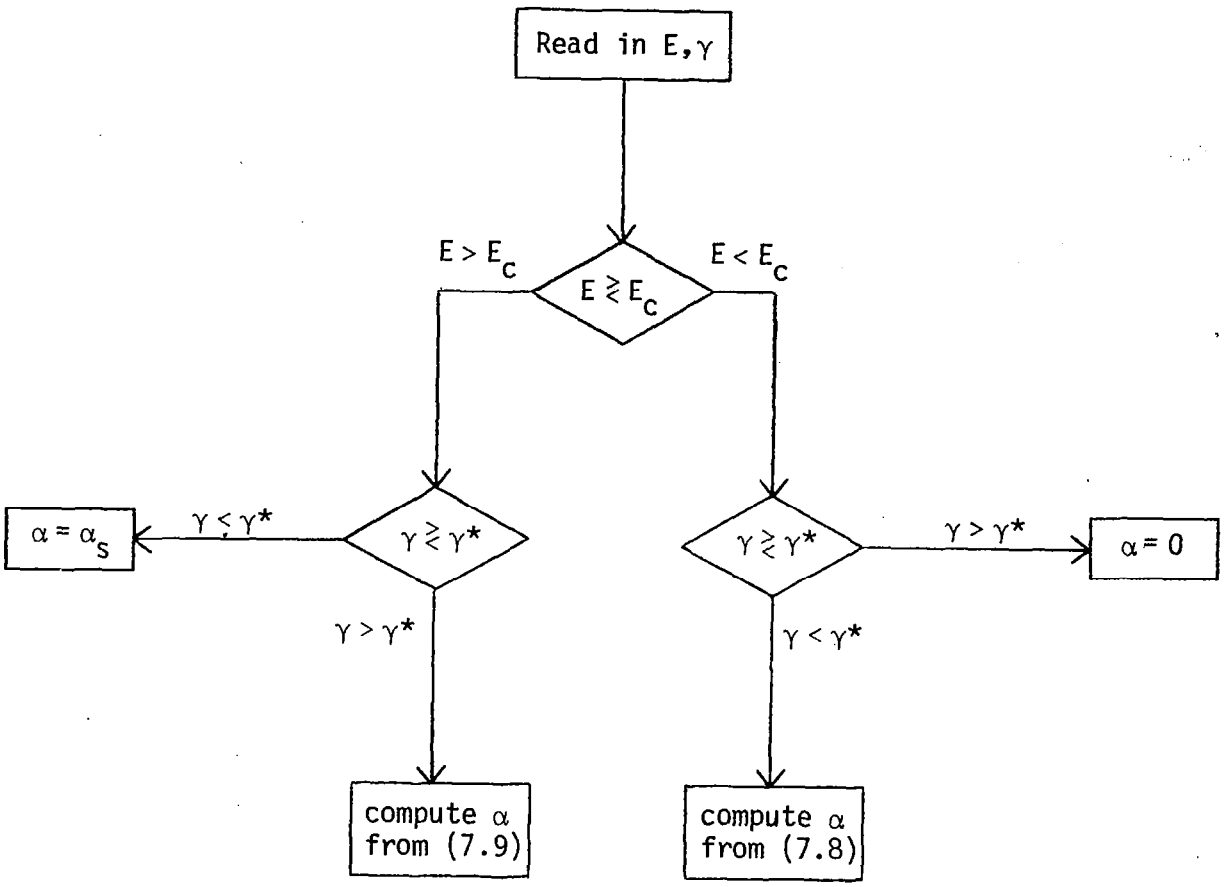


Figure 5.7.1

Feedback Control Law $\alpha(\gamma-\gamma^*)$ in the γ-Boundary Layer

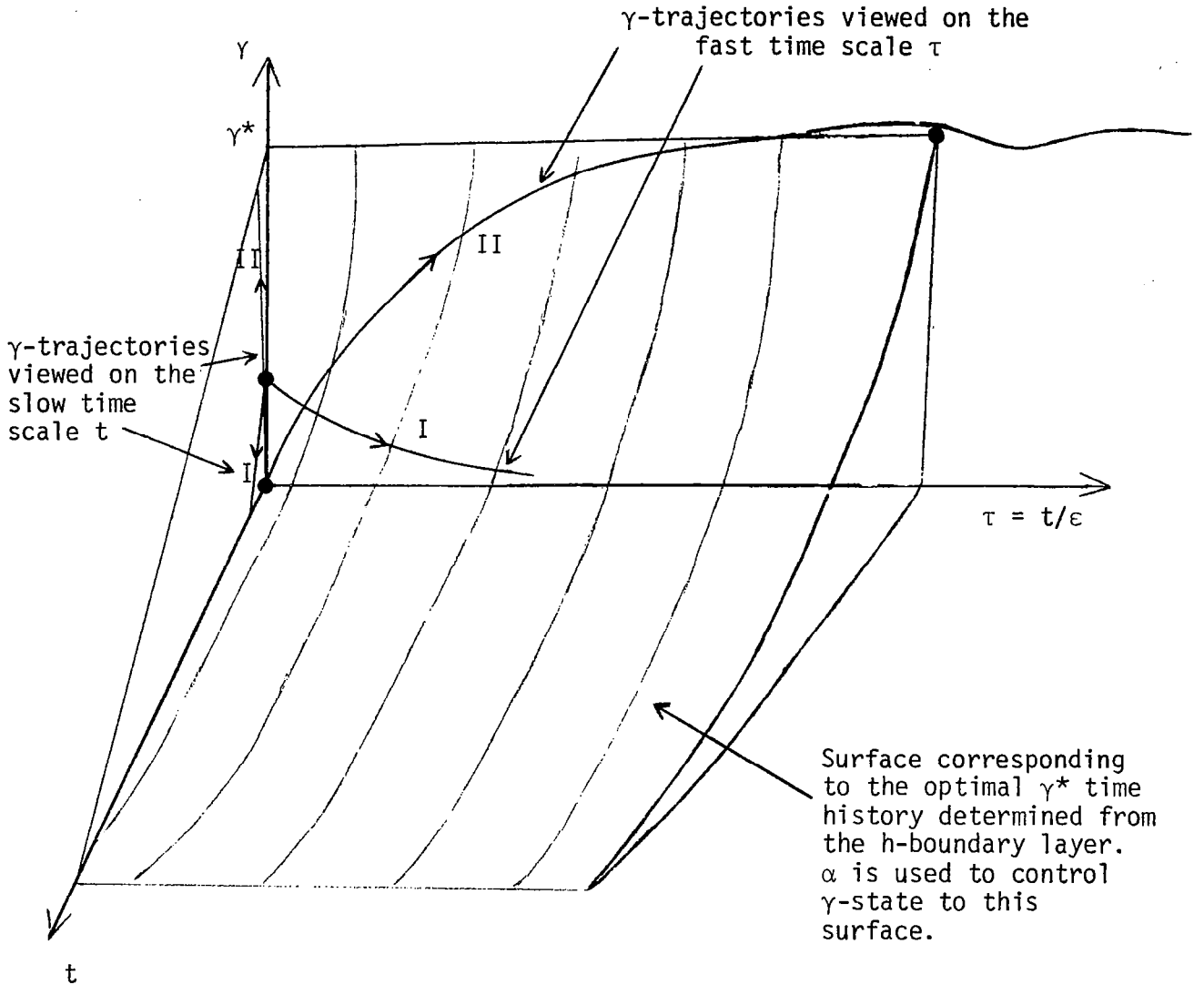


Figure 5.7.3

γ-Boundary Layer Control

The γ-trajectories are shown on two time scales, t and $\tau = t/\varepsilon$. The t time scale corresponds to events in the h-boundary layer and the change in γ on this time scale appears instantaneous. Case I shows a trajectory for $\gamma^* = 0$ and $\gamma > \gamma^*$, $E < E_c$. From Figure 5.7.1, the optimal $\alpha = 0$ till $\gamma = \gamma^*$ and α = [equation (7.8)] when $\gamma < \gamma^*$. The asymptotic value of α is determined from the α-value on the h-boundary layer, namely $\alpha \simeq \frac{W}{T+L_\alpha}$. Notice that a slight undershoot in γ would occur and some form of anticipatory action may be used to reduce this undershoot and consequent chatter. Case II corresponds to an asymptotic value $\gamma = \gamma^*$, $\gamma(\tau = 0) < \gamma^*$ and $E > E_c$. For this case, Figure 5.7.1 shows that the optimal value is α_s till $\gamma(\tau) > \gamma^*$ and α = [equation (7.9)] for $\gamma < \gamma^*$. In practice, it would be better to set α to its asymptotic value once $|\gamma - \gamma^*|$ is less than a small threshold value. An alternative would be to switch to linearized control once $|\gamma - \gamma^*|$ decreases below a prespecified value. A similar logic should also be used for other boundary layers.

APPENDIX 5.1
DERIVATION OF THE EXACT E,β SOLUTION

A5.1.1 Introduction

As we saw in Subsection 5.5.1 of Chapter 5, the equations of motion for the x,y,E,β system (referred to as the energy-state approximation in Parsons (1972)) are

$$(1.1) \qquad \dot{E} = \frac{V(T - D_0 - D_L \sec^2\sigma)}{W}$$

$$(1.2) \qquad \dot{\beta} = \frac{g \tan\sigma}{V}$$

$$(1.3) \qquad \dot{x} = V \cos\beta$$

$$(1.4) \qquad \dot{y} = V \sin\beta$$

E, β, x and y are the four states and T, V and σ are the three controls. The three states V, h and γ of the original dynamic equations have been reduced to the single state E and V or h becomes a control variable. Essentially, the assumption introduced here is that V and h can be changed instantaneously at constant E by zoom climbs or zoom dives. For consistency with equations (1.3) and (1.4) these maneuvers also occur without changes in x or y. Section 5.6 of Chapter 5 presents the concept of boundary layer corrections to smooth out zoom climbs or dives as in Figure A5.1.1.

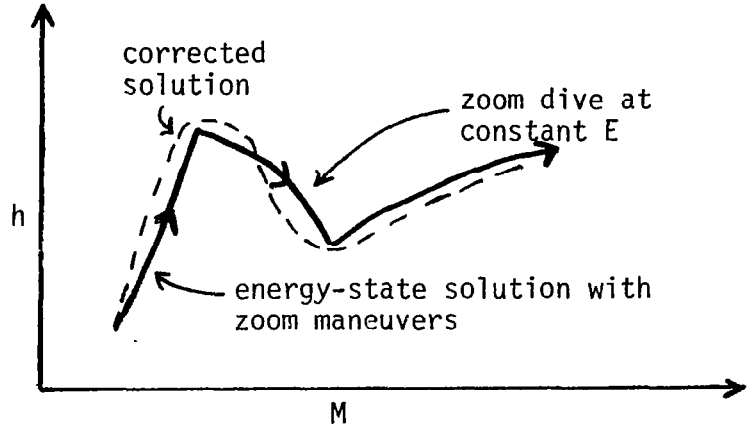


Figure A5.1.1.-Concept of Boundary Layer Corrections.

The major assumptions implied in the equations of motion (1.1) to (1.4) are as follows: (a) change of mass is negligible, (b) cosine of angle-of-attack is approximately one, (c) *the normal component of thrust is negligible compared to lift.*

The boundary conditions that go with the formulation of (1.1) through (1.4) are for a turn to a point:

(1.5) $E(0) = E_0$ $\qquad$ $E(t_f) = E_f$

(1.6) $\beta(0) = \beta_0$ $\qquad$ $\beta(t_f) = \beta_f$

(1.7) $x(0) = x_0$ $\qquad$ $x(t_f) = x_f$

(1.8) $y(0) = y_0$ $\qquad$ $y(t_f) = y_f$

and for a turn to a line:

(1.9) $\beta(t_f) = \pi/2$ $\qquad$ (rotate x-axis so that it is normal to objective line)

(1.10) $x(t_f) = x_f$

(1.11) $y(t_f) = \text{free}$

The constraints on the three controls are:

(1.12) $0 = T_{min} \leq T \leq T_{max}(h,M)$

(1.13) $|\alpha| \leq \alpha_s(V,E,\alpha_s)$

(1.14) $V \leq V_{max}(E)$

(1.15) $|\sigma| \leq \sigma_m$

The minimum or idle thrust is assumed to be zero. σ_s is the bank angle which yields the stall angle-of-attack $\alpha_s = 12°$ at an existing energy and velocity from the equation:

(1.16) $$\alpha_s = \cos^{-1}\left[\frac{W}{L_\alpha \alpha_s}\right]$$

The V_{max} constraint combines an approximate maximum air speed limit, a maximum Mach number contraint, and a positive altitude constraint. Figure A5.1.2 shows the form of this constraint. The maximum bank angle constraint σ_m is a maximum normal load constraint:

(1.17) normal load in g's $= \frac{V\dot{\beta}}{g} = \tan\sigma$
assumed to be 4. Thus $\sigma_m = 76°$.

The auxiliary equations that relate all the variables present in the equations of motion (1.1) to (1.4) are

$$h = E - \frac{V^2}{2g} \tag{1.18}$$

$$\alpha = \frac{W \sec\sigma}{L_\alpha} \tag{1.19}$$

by using the assumptions $\cos\gamma \simeq 1$ and $\dot{\gamma} \simeq 0$,

$$L = \alpha L_\alpha \tag{1.20}$$

$$D = D_0 + D_L \sec^2\sigma \tag{1.21}$$

$$q \triangleq \tfrac{1}{2}\rho V^2 \tag{1.22}$$

$$L_\alpha \triangleq C_{L_\alpha} qS \tag{1.23}$$

$$D_0 \triangleq C_{D_0} qS \tag{1.24}$$

$$D_L \triangleq \frac{\eta W^2}{L_\alpha} \tag{1.25}$$

and the following constants:

(1.26) $S = 49.239\ m^2$
$W = 1.5569 \times 10^5\ N$

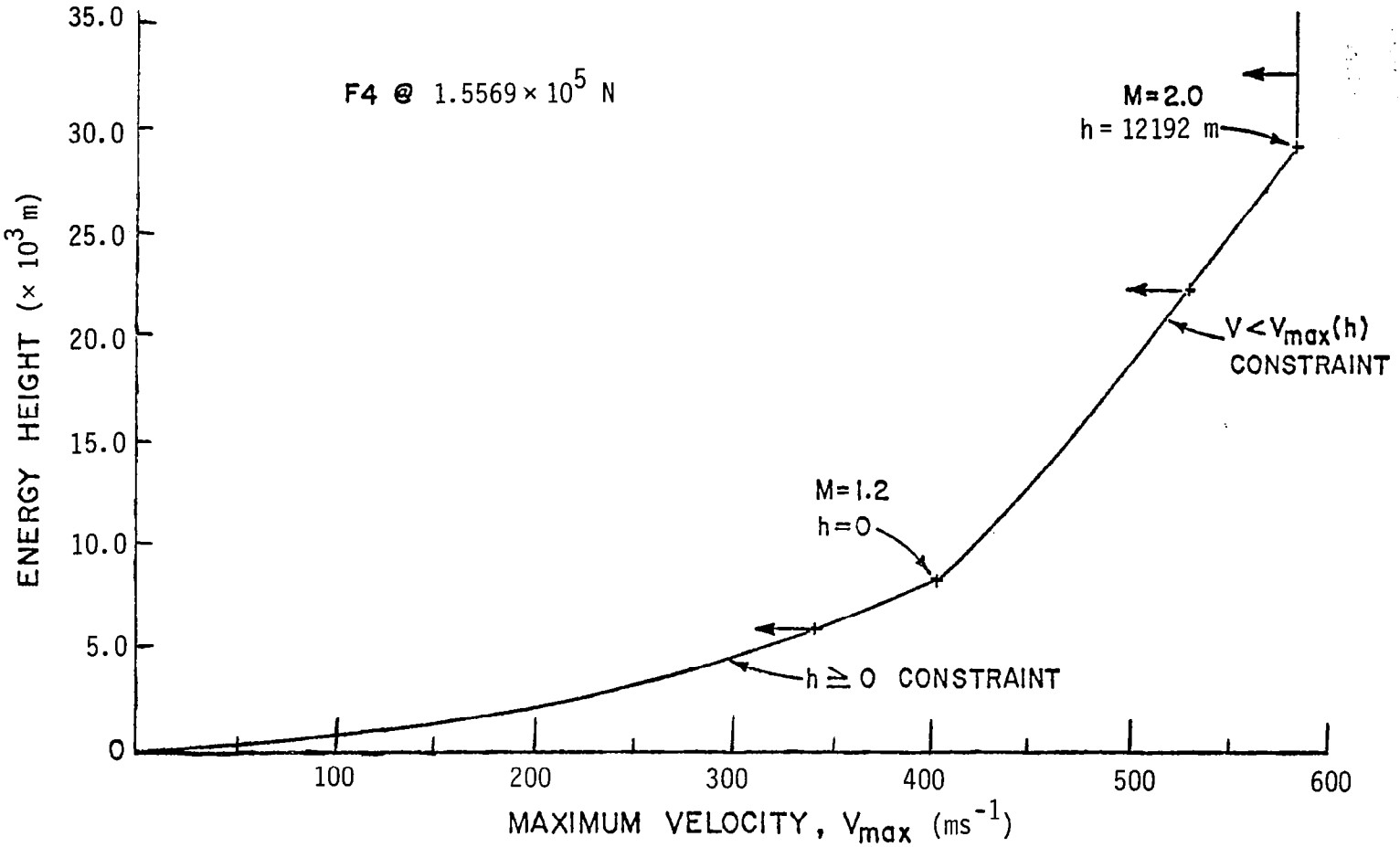


Figure A5.1.2.-V_{max} Constraint as a Function of Energy-Height.

The example aircraft being used in this work is an early version of the F-4. Sonic speed $c(h)$, density $\rho(h)$, lift coefficient slope $C_{L_\alpha}(M)$, zero lift drag coefficient $C_{D_0}(M)$, aerodynamic efficiency factor $\eta(M)$ and maximum thrust $T_{max}(h,M)$ are tabulated functions. The numerical treatment of this data so as to provide continuous values by construction of spline fits was described in Chapter 2 and so will not be discussed further here.

A5.1.2 First-Order Necessary Conditions

For this problem the Hamiltonian may be written as

$$(1.27)\quad H = 1 + \frac{\lambda_E V}{W}(T - D_0 - D_L \sec^2\sigma) + \lambda_\beta g \frac{\tan\sigma}{V} + \lambda_x V \cos\beta + \lambda_y V \sin\beta$$
$$+ \mu_1 T(T - T_{max}) + \mu_2\left(\frac{W \sec\sigma}{L_\alpha} - \alpha_s\right) + \mu_3(V - V_{max}) + \mu_4(\tan^2\sigma - \tan^2\sigma_m)$$

where

$$(1.28)\quad \begin{aligned}
&\mu_1 = 0 && \text{if} && 0 < T < T_{max}\\
&\mu_1 \geq 0 && \text{if} && T = 0 \text{ or } T = T_{max}\\
&\mu_2 = 0 && \text{if} && |\sigma| < \sigma_s\\
&\mu_2 \geq 0 && \text{if} && |\sigma| = \sigma_s\\
&\mu_3 = 0 && \text{if} && V < V_{max}\\
&\mu_3 \geq 0 && \text{if} && V = V_{max}\\
&\mu_4 = 0 && \text{if} && |\sigma| < \sigma_m\\
&\mu_4 \geq 0 && \text{if} && |\sigma| = \sigma_m
\end{aligned}$$

Euler-Lagrange Equations

$$(1.29)\quad \dot{\lambda}_E = -\frac{\partial H}{\partial E}$$
$$= \frac{\lambda_E V}{W}\left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E}\sec^2\sigma\right) + \mu_1 T \frac{\partial T_{max}}{\partial E} + \mu_2 \frac{W \sec\sigma}{L_\alpha^2}\frac{\partial L_\alpha}{\partial E} + \mu_3 \frac{\partial V_{max}}{\partial E}$$

(1.30) $$\dot{\lambda}_\beta = -\frac{\partial H}{\partial \beta}$$

$$= \lambda_x V \sin\beta - \lambda_y V \cos\beta$$

(1.31) $$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = 0$$

(1.32) $$\dot{\lambda}_y = -\frac{\partial H}{\partial y} = 0$$

Hence both λ_x and λ_y are constant.

Optimality Conditions

(1.33) $$\frac{\partial H}{\partial T} = 0$$

$$= \frac{\lambda_E V}{W} + \mu_1(2T - T_{max})$$

(1.34) $$\frac{\partial H}{\partial \sigma} = 0$$

$$= \sec^2\sigma \left[\frac{g\lambda_\beta}{V} - \frac{2D_L V \lambda_E \tan\sigma}{W} + \mu_2 \frac{W \sec\sigma}{L_\alpha} + 2\mu_4 \tan\sigma\right]$$

(1.35) $$\frac{\partial H}{\partial V} = 0 = \frac{\lambda_E}{W}(T - D_0 - D_L \sec^2\sigma - V\frac{\partial D_0}{\partial V} - V\frac{\partial D_L}{\partial V}\sec^2\sigma) - \lambda_\beta g \frac{\tan\sigma}{V^2} + \lambda_x \cos\beta$$

$$+ \lambda_y \sin\beta - \mu_1 T \frac{\partial T_{max}}{\partial V} - \mu_2 \frac{W \sec\sigma}{L_\alpha^2}\frac{\partial L_\alpha}{\partial V} + \mu_3$$

Transversality Condition

(1.36) $$H(t_f) = 0$$

Also since the Hamiltonian is implicit in the independent variable, time, we have

(1.37) $$H(t) = H(t_f) = 0$$

Equations (1.1) through (1.8), (1.28) through (1.35) and (1.37) are the first-order necessary conditions for a minimum-time turn.

A5.1.3 Legendre-Clebsch Necessary Conditions for a Minimum (2^{nd} Order Conditions)

The Legendre-Clebsch necessary condition for a minimum is

$$(1.38) \qquad H_{uu} = \begin{bmatrix} H_{TT} & H_{\sigma T} & H_{VT} \\ H_{T\sigma} & H_{\sigma\sigma} & H_{V\sigma} \\ H_{TV} & H_{\sigma V} & H_{VV} \end{bmatrix} \geq 0$$

if $\det H_{uu} = 0$, then the time optimal trajectory is a singular arc.

The components of H_{uu} have the following form:

$$(1.39) \qquad H_{TT} = 2\mu_1$$

$$(1.40) \qquad H_{\sigma T} = H_{T\sigma} = 0$$

$$(1.41) \qquad H_{VT} = H_{TV} = \frac{\lambda_E}{V} - \mu_1 \frac{\partial T_{max}}{\partial V}$$

$$(1.42) \qquad H_{\sigma\sigma} = \sec^4\sigma \left[- \frac{2D_L V \lambda_E}{W} + \mu_2 \frac{W \cos^3\sigma}{L_\alpha} + 2\mu_4 \right]$$

$$(1.43) \qquad H_{V\sigma} = H_{\sigma V} = \sec^2\sigma \left[- \frac{g\lambda_\beta}{V^2} - \frac{2D_L \lambda_E \tan\sigma}{W} - 2 \frac{\frac{\partial D_L}{\partial V} V\lambda_E \tan\sigma}{W} - \mu_2 W \frac{\partial L_\alpha}{\partial V} \frac{\sin\sigma}{L_\alpha^2} \right]$$

$$(1.44) \qquad H_{VV} = \frac{2\lambda_\beta g \tan\sigma}{V^3} - \frac{\lambda_E}{W} \left[2 \frac{\partial D_0}{\partial V} + 2 \frac{\partial D_L}{\partial V} \sec^2\sigma + V \frac{\partial^2 D_0}{\partial V^2} + V \frac{\partial^2 D_L}{\partial V^2} \sec^2\sigma \right]$$

$$- \mu_1 T \frac{\partial^2 T_{max}}{\partial V^2} + \mu_2 \frac{W \sec\sigma}{L_\alpha^2} \left[\frac{2}{L_\alpha} \left(\frac{\partial L_\alpha}{\partial V} \right)^2 - \frac{\partial^2 L_\alpha}{\partial V^2} \right]$$

A5.1.4 Existence of a Maximum Mach Number Straight Cruise Arc

Parsons (1972) gives a full justification for the existence of a singular arc with straight flight using intermediate thrust and maximum Mach number using

equations (1.29) through (1.35) and (1.38) through (1.44). We will only present the characteristics of this cruise arc here.

The cruise energy and velocity are given by the relation

$$(1.45) \qquad \frac{\partial V_{max}}{\partial E} = 0$$

Since the maximum velocity constraint is of the form shown in Figure A5.1.2, the energy at which (1.45) is satisfied is unspecified at or above 29.949 Km. As it is expected that initial and final energies will be below the minimum cruise energy, this value is chosen as the cruise energy. Hence we have

$$(1.46) \qquad E_c = \min[E]\Big|_{\frac{\partial V_{max}}{\partial E} = 0} = 2.9949 \times 10^4 \text{ m}$$

$$(1.47) \qquad V^* = \hat{V}_{max} = V_{max}(E)\Big|_{\frac{\partial V_{max}}{\partial E} = 0} = 590.2 \text{ ms}^{-1}$$

$$(1.48) \qquad \sigma^* = 0$$

$$(1.49) \qquad T^* = T_c = D_0(\hat{V}_{max}, E_c) + D_L(\hat{V}_{max}, E_c)$$

$$(1.50) \qquad h_c = 1.2192 \times 10^4 \text{ m}$$

Equations (1.46) to (1.50) characterize the cruise arc and so at present we only consider a three-dimensional turn to a point or onto a line which is far enough away from the starting point so that a cruise arc is reached during the flight. The length of the cruise arc is determined so that the horizontal plane boundary conditions are satisfied. Thus, the initial turn to cruise arc and the final turn from the cruise arc can be determined separately. The solution therefore consists of three parts: (1) initial turn to cruise arc, (2) flight along cruise arc, (3) final turn to required point from cruise arc.

A5.1.5 Formulation of the Initial Turn

As the coordinate system can be rotated so that the cruise heading, $\beta_s = 0°$,

we only consider an initial turn onto the line $y = 0$ within the cruise arc as in Figure A5.1.3.

In order to facilitate the formulation of the initial turn we assume that the turn contains a constant radius turning segment with $T^* = T_{max}$, $E = E_c$ and $V^* = \hat{V}_{max}$ as in Figure A5.1.4.

The cost function, J, for minimum time to $x = 0$ is

$$(1.51) \quad \min_{T,\sigma,V} \left\{ J = \underbrace{\frac{-x_1}{\hat{V}_{max}} + \frac{R_{min}\sin\beta_1}{\hat{V}_{max}}}_{\text{time in cruise}} - \underbrace{\frac{R_{min}\beta_1}{\hat{V}_{max}}}_{\substack{\text{time in turn at} \\ V = \hat{V}_{max}}} + \underbrace{\int_{t_0}^{t_1} dt}_{\substack{\text{time in turn at} \\ V < \hat{V}_{max}}} \right\}$$

subject to the equations of motion (1.1) to (1.4), and the boundary conditions are now

(1.52) $E(t_0) = E_0$ $\qquad$ $E(t_1) = E_c$

(1.53) $\beta(t_0) = \beta_0 < 0$ $\qquad$ $\beta(t_1) = \beta_1$, free parameter

(1.54) $x(t_0) = x_0$ $\qquad$ $x(t_1) = R_{min}\sin\beta_1$, free (strictly, $x(t_1) = R_{min}\sin\beta_1 + x_f$, but without loss of generality $x_f = 0$.)

(1.55) $y(t_0) = y_0$ $\qquad$ $y(t_1) = R_{min}(1 - \cos\beta_1)$, free

(1.54) and (1.55) are obtained from the geometry of the problem.

A5.1.6 Determination of the Optimal Initial Turn Trajectory

The variational Hamiltonian is the same as equation (1.27) and first-order necessary conditions and first integral as in equations (1.29) to (1.37). The following terminal adjoints may be obtained from (1.51):

(1.56) $\lambda_E(t_1) =$ unknown

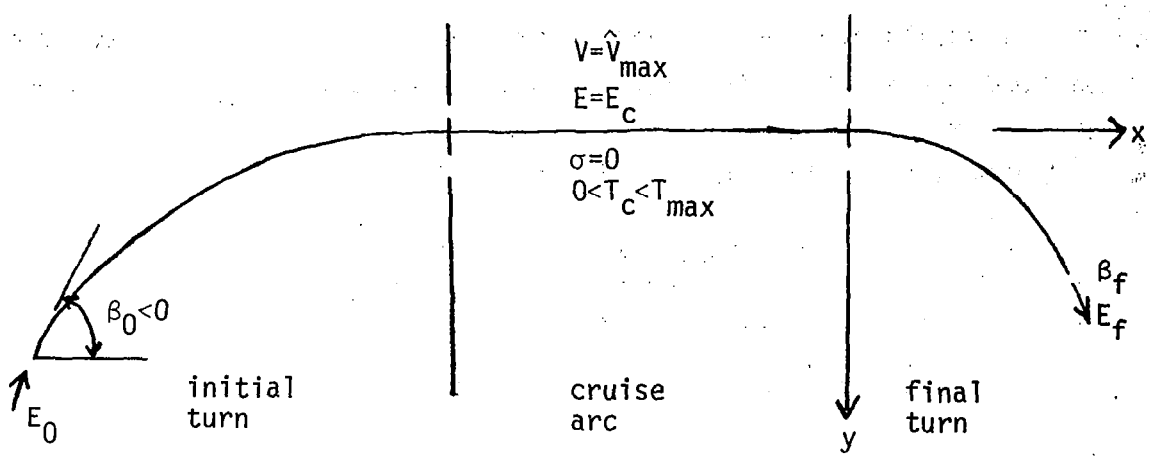


Figure A5.1.3.-Break-up of Trajectory: Initial Turn, Cruise Arc, Final Turn.

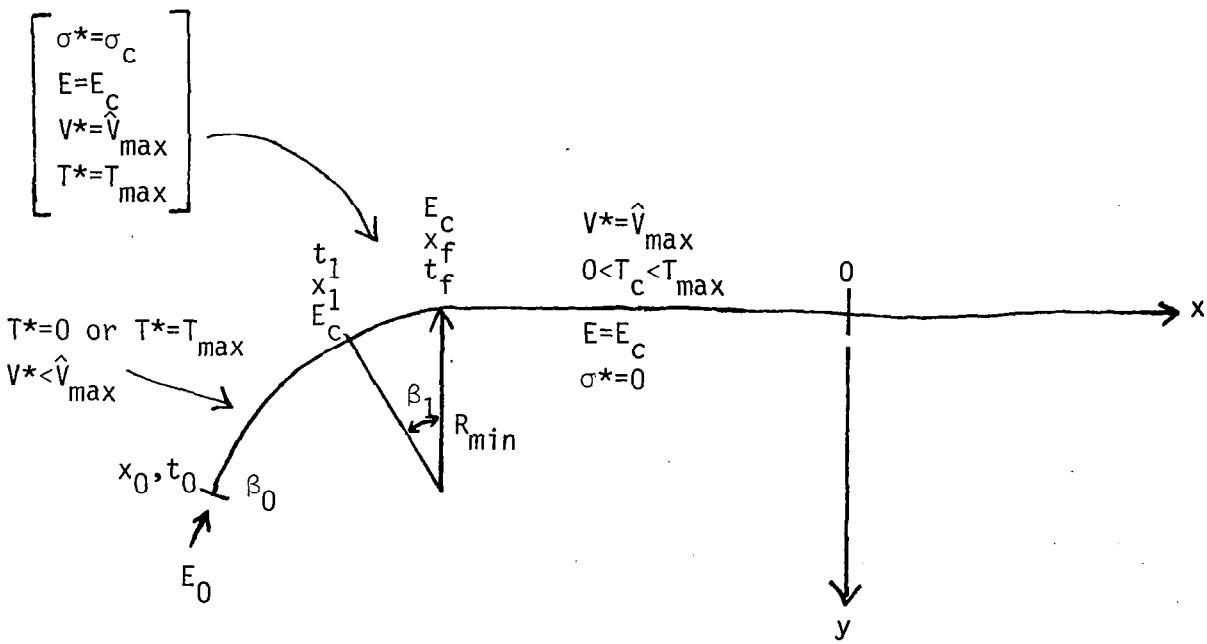


Figure A5.1.4.-Constant Radius Turning Segment Within Initial Turn.

$$(1.57) \qquad \lambda_\beta(t_1) = \frac{-R_{min}}{\hat{V}_{max}} (1 - \cos\beta_1)$$

$$(1.58) \qquad \lambda_x(t_1) = \frac{-1}{\hat{V}_{max}}$$

$$(1.59) \qquad \lambda_y(t_1) = 0$$

With the results (1.58) and (1.59), equation (1.30) becomes

$$(1.60) \qquad \dot{\lambda}_\beta = \frac{-V \sin\beta}{\hat{V}_{max}}$$

It is intended to integrate the state and adjoint equations backwards from time t_1, hence determination of $\lambda_E(t_1)$ together with the values of the controls at t_1 will parameterize the whole problem in terms of β_1. The value of t_1 is also in terms of β_1:

$$(1.61) \qquad t_1 = \left.\frac{-\beta_1}{\dot{\beta}}\right|_{\substack{V = \hat{V}_{max} \\ E = E_c}}$$

Controls at Time t_1

On the constant radius segment, bank angle for constant energy will be

$$(1.62) \qquad \sigma_c = \tan^{-1}\left[\left.\frac{T_{max} - D_0 - D_L}{D_L}\right|_{\substack{V = \hat{V}_{max} \\ E = E_c}}\right]^{1/2} = 59.2°$$

since $\dot{E} = 0$

$$(1.63) \qquad R_{min} = \left.\frac{\hat{V}_{max}}{\dot{\beta}}\right|_{\substack{V = \hat{V}_{max} \\ E = E_c}} = \frac{\hat{V}_{max}^2}{g \tan\sigma_c} = 2.1195 \times 10^4 \text{ m}$$

(1.64) $$T = T_{max}(\hat{V}_{max}, E_c) = 10.45 \times 10^4 \text{ N}$$

(1.65) $$V = \hat{V}_{max} = 590.2 \text{ ms}^{-1}$$

Determination of $\lambda_E(t_1)$

Using (1.34) when $|\sigma| < \sigma_s$ and $|\sigma| < \sigma_m$ so that $\mu_1 = \mu_4 = 0$ and $\sigma = \sigma_u$ = optimal unconstrained bank angle,

(1.66) $$\lambda_E = \frac{Wg\lambda_\beta}{2D_L V^2 \tan\sigma_u}$$

for continuity,

(1.67) $$\tan\sigma_u \Big|_{t_1^-} = \tan\sigma_c$$

(1.68) $$V\Big|_{t_1^-} = \hat{V}_{max}$$

Using (1.57), (1.63), (1.66) to (1.68) we have

(1.69) $$\lambda_E(t_1) = \frac{-W(1 - \cos\beta_1)}{2D_L V_{max} \tan^2\sigma_c}$$

Thus the initial turn problem has only the single parameter β_1 to be selected to satisfy a given E_0 at β_0 or vice versa.

Choice of Optimal Controls Along Trajectory

The optimal bank angle is obtained from

(1.70) $$\sigma^* = \min [\sigma_u, \sigma_s, \sigma_m]$$

where σ_m is given by equation (1.17) and σ_s by equation (1.16). The unconstrained optimal bank angle, σ_u, is obtained from:

(1.71) $$\sigma_u = \tan^{-1}\left[\frac{Wg\lambda_\beta}{2D_L V^2 \lambda_E}\right]$$

Equations (1.28) and (1.33) require that:

(1.72) $\lambda_E > 0$ if $T = 0$ since $\mu_1 = \dfrac{\lambda_E V}{WT_{max}} \geq 0$

(1.73) $\lambda_E = 0$ if $0 < T < T_{max}$

(1.74) $\lambda_E < 0$ if $T = T_{max}$ since $\mu_1 = \dfrac{-\lambda_E V}{WT_{max}} \geq 0$

Hence an observation of the sign of λ_E determines the optimal thrust. A more complete derivation of σ^* and T^* appears in Parsons (1972).

The optimal velocity could be obtained from the remaining optimality condition equation (1.35) with equation (1.28). This requires a one-dimensional search in V at constant state from the V_{max} constraint to a value around the stall velocity. However, to eliminate the need to calculate the partial derivatives of equation (1.35) in the search, the optimum velocity was obtained by minimizing the variable portion of the Hamiltonian:

$$(1.75) \qquad V^* = \underset{V_{LM}<V<V_{max}}{\arg\min} \left[\frac{\lambda_E W}{W} (T^* - D_0 - D_L \sec^2\sigma^*) + \lambda_\beta \frac{g \tan\sigma^*}{V} - \frac{V \cos\sigma^*}{\hat{V}_{max}} \right]\Bigg|_{\substack{\text{constant} \\ \text{state}}}$$

where V_{LM} is the lower limit of the search.

$\dot{\lambda}_E$ Equations

Using $\lambda_x = -\dfrac{1}{\hat{V}_{max}}$ and $\lambda_y = 0$ and equations (1.33) to (1.35) to get the $\mu(t)$, equation (1.29) for $\dot{\lambda}_E$ becomes:

$$(1.76) \qquad \dot{\lambda}_E = \frac{\lambda_E W}{W} \left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E} \sec^2\sigma - \xi \frac{\partial T_{max}}{\partial E} \right)$$

when $|\sigma| < \sigma_s$ and $V < V_{max}$

$$(1.77)\qquad \dot{\lambda}_E = \frac{\lambda_E W}{W}\left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E}\sec^2\sigma - \xi\frac{\partial T_{max}}{\partial E}\right)$$

$$+ \frac{\partial V_{max}}{\partial E}\left\{\frac{\cos\beta}{\hat{V}_{max}} - \frac{\lambda_E}{W}\left[\xi\left(T + V\frac{\partial T_{max}}{\partial V}\right) - \left(D_0 + V\frac{\partial D_0}{\partial V}\right)\right.\right.$$

$$\left.\left. - \left(D_L + V\frac{\partial D_L}{\partial V}\right)\sec^2\sigma\right] + \frac{\lambda_\beta g\tan\sigma}{V^2}\right\}$$

when $|\sigma| < \sigma_s$ and $V = V_{max}$

$$(1.78)\qquad \dot{\lambda}_E = \frac{\lambda_E V}{W}\left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E}\sec^2\sigma - \xi\frac{\partial T_{max}}{\partial E}\right) + \frac{\partial L_\alpha}{\partial E}\left(\frac{2D_L V\lambda_E}{WL_\alpha\cos^2\sigma} - \frac{g\lambda_\beta}{VL_\alpha\sin\sigma\cos\sigma}\right)$$

when $|\sigma| = \sigma_s < \sigma_m$ and $V < V_{max}$

$$(1.79)\qquad \dot{\lambda}_E = \frac{\lambda_E V}{W}\left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E}\sec^2\sigma - \xi\frac{\partial T_{max}}{\partial E}\right) + \frac{\partial V_{max}}{\partial E}\left\{\frac{\cos\beta}{\hat{V}_{max}}\right.$$

$$- \frac{\lambda_E}{W}\left[\xi\left(T + V\frac{\partial T_{max}}{\partial V}\right) - \left(D_0 + V\frac{\partial D_0}{\partial V}\right) - \left(D_L + V\frac{\partial D_L}{\partial V}\right)\sec^2\sigma\right] + \frac{\lambda_\beta g\tan\sigma}{V^2}$$

$$\left. + \frac{\partial L_\alpha}{\partial V}\left(\frac{2D_L V\lambda_E}{WL_\alpha\cos^2\sigma} - \frac{g\lambda_\beta}{VL_\alpha\sin\sigma\cos\sigma}\right)\right\} + \frac{\partial L_\alpha}{\partial E}\left[\frac{2D_L V\lambda_E}{WL_\alpha\cos^2\sigma} - \frac{g\lambda_\beta}{VL_\alpha\sin\sigma\cos\sigma}\right]$$

when $|\sigma| = \sigma_s$ and $V = V_{max}$

$$(1.80)\quad \dot{\lambda}_E = \frac{\lambda_E V}{W}\left(\frac{\partial D_0}{\partial E} + \frac{\partial D_L}{\partial E}\sec^2\sigma - \xi\frac{\partial T_{max}}{\partial E}\right) - \frac{\frac{\partial L_\alpha}{\partial E}}{\frac{\partial L_\alpha}{\partial V}}\left\{\frac{\cos\beta}{\hat{V}_{max}} - \frac{\lambda_E}{W}\left[\xi\left(T + V\frac{\partial T_{max}}{\partial V}\right) - \left(D_0 + V\frac{\partial D_0}{\partial V}\right) - \left(D_L + V\frac{\partial D_L}{\partial V}\right)\sec^2\sigma\right] + \frac{\lambda_\beta g\tan\sigma}{V^2}\right\}$$

when $|\sigma| = \sigma_s = \sigma_m$ and $V < V_{max}$ and where $\xi = 1$ if $T = T_{max}$ and $\xi = 0$ if $T = 0$.

This completes the formulation of the initial turn problem with the single parameter β_1. A choice of β_1 specifies all the initial conditions for the backward integration of the $\dot{E}$, $\dot{\beta}$, $\dot{x}$, $\dot{y}$, $\dot{\lambda}_E$ and $\dot{\lambda}_\beta$ equations to produce an optimal trajectory to satisfy a given E_0 at β_0 or vice versa.

A5.1.7 Alternate Single Parameter Formulation

The formulation of Section A5.1.6 shows that $\sigma = \sigma_c = 59.2°$ at $t = t_1$. The optimum bank angle is almost zero until t is very close to t_1, thus producing an extremely sharp spike in σ. This is very pronounced for $|\beta_1| < 1° \times 10^{-6}$. As in Figure A5.1.5, since digital computations are being used, σ is held constant over each integration step, thus broadening this spike in bank angle. To attempt to follow the sharp change in σ more accurately, the step length must be reduced considerably, resulting in increased computation time. Thus, to avoid these numerical difficulties, an alternate single parameter formulation may be obtained using $\lambda_\beta(1)$ as the parameter and integrating backwards from $t = 1$ instead of $t = t_1$. A review of the results for $0 < t \leq 1$ for small $|\beta_1|$ revealed that λ_E, E and x are all independent of λ_β and β. Hence the values of $\lambda_E(1)$, $E(1)$, and $x(1)$ for $|\beta_1| = 10^{-6}$ can be used as initial values to begin integration at $t = 1$ for trajectories given by $|\beta_1| < 10^{-6}$. These values are:

$$(1.81)\quad \lambda_E(1) = -1.52 \times 10^{-5}\ m^{-1}s$$

$$(1.82)\quad E(1) = 2.9907 \times 10^{4}\ m$$

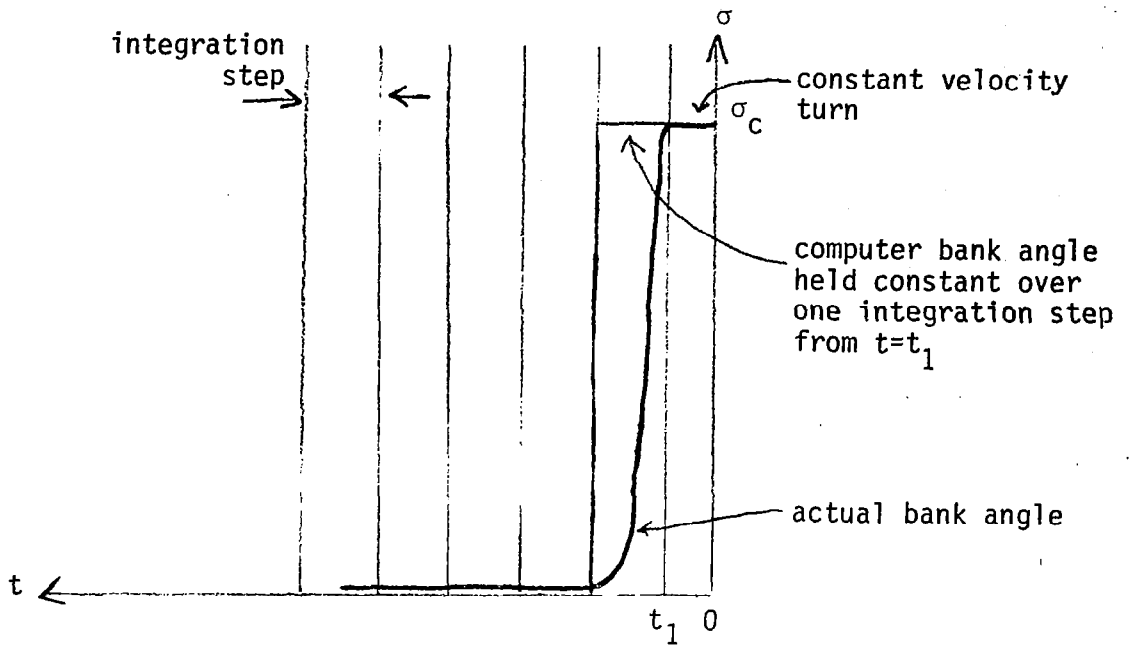


Figure A5.1.5.-Numerical Problems in Following Spike in Bank Angle.

(1.83) $x(1) = -0.0278 \times R_{min}$ m

The controls at $t = 1$ can be taken as

(1.84) $V^*(1) = \hat{V}_{max}$

(1.85) $\sigma^*(1) = 0$

(1.86) $T^*(1) = T_{max}$

Further, a study of the solution near $t = t_1$ from Parsons (1972) gives an approximate value for $\beta(1)$ as:

(1.87) $\beta(1) = 0.53\ \lambda_\beta(1)$

Thus we now have the initial turn problem parameterized by $\lambda_\beta(1)$ to be selected to satisfy E_0 at β_0 or vice versa.

A5.1.8 Numerical Results for Initial Arc

Several optimal trajectories were obtained by variation of the parameter β_1 or $\lambda_\beta(1)$. Parsons (1972) shows that the complete set of initial conditions is covered by the range of parameter values $0 < \beta_1 < -2.15°$. For values of $|\beta_1|$ above about 10^{-3}, the entire initial turn is on the maximum velocity constraint. For $|\beta_1| < 10^{-6}$ the $\lambda_\beta(1)$ formulation was used and we present here the characteristics of the optimal trajectory obtained for a choice of $\lambda_\beta(1) = -10^{-17}$ which displays the zoom dives that reflect the energy-state approximation used in the solution.

Figure A5.1.6 shows the flowchart for obtaining an optimal trajectory backwards in time from the cruise arc. Box A and the dotted feedback loop indicate logic to choose the correct value of the parameter β_1 or $\lambda_\beta(1)$ to arrive at specified initial conditions E_0 and β_0. This logic would interpolate between the family of β_1 or $\lambda_\beta(1)$ trajectories (flooding method). This may involve a few iterations on those parts of the trajectory where β changes.

For the example trajectory chosen ($\lambda_\beta(1) = -10^{-17}$), the stopping criterion used was to test if the $|\sigma| = \sigma_m = \sigma_s$ locus had been reached. This is the point where the angle-of-attack, α, and the bank angle are at their maximum values simultaneously. Figures A5.1.11 - A5.1.23 give the time-histories for all the states and controls and for some of the derived variables.* The changes in velocity and altitude appear in Figures A5.1.13 and A5.1.14 where the zoom dives clearly indicate the instantaneous tradeoff between height and velocity. Figure A5.1.15 reveals that the whole initial turn consisting of 104° is completed within the first minute of the trajectory. The change in σ (Figure A5.1.16) is also completed in the first minute corresponding to the change in heading angle. The spike in σ introduced by the constant velocity turn upon reaching the cruise arc may be ignored in practice with no effect upon the optimal trajectory. Note that the entire initial turn is accomplished using $T^* = T_{max}$: the changes appearing in Figure A5.1.17 are due entirely to the changes in height and Mach number upon which T_{max} depends. As expected from Figure A5.1.4 where the initial turn was formulated, the necessary adjustment in the y coordinate is completed in the first minute (Figure A5.1.19) during the change in heading angle. Figure A5.1.21 shows the time-history of the derived variable α. Notice that $\alpha = 12°$ at the beginning of the turn where $|\sigma| = \sigma_m = \sigma_s$ which is the max turn locus. Figure A5.1.24

*The Figures A5.1.11 - A5.1.42 are collected at the end of Appendix 5.1.

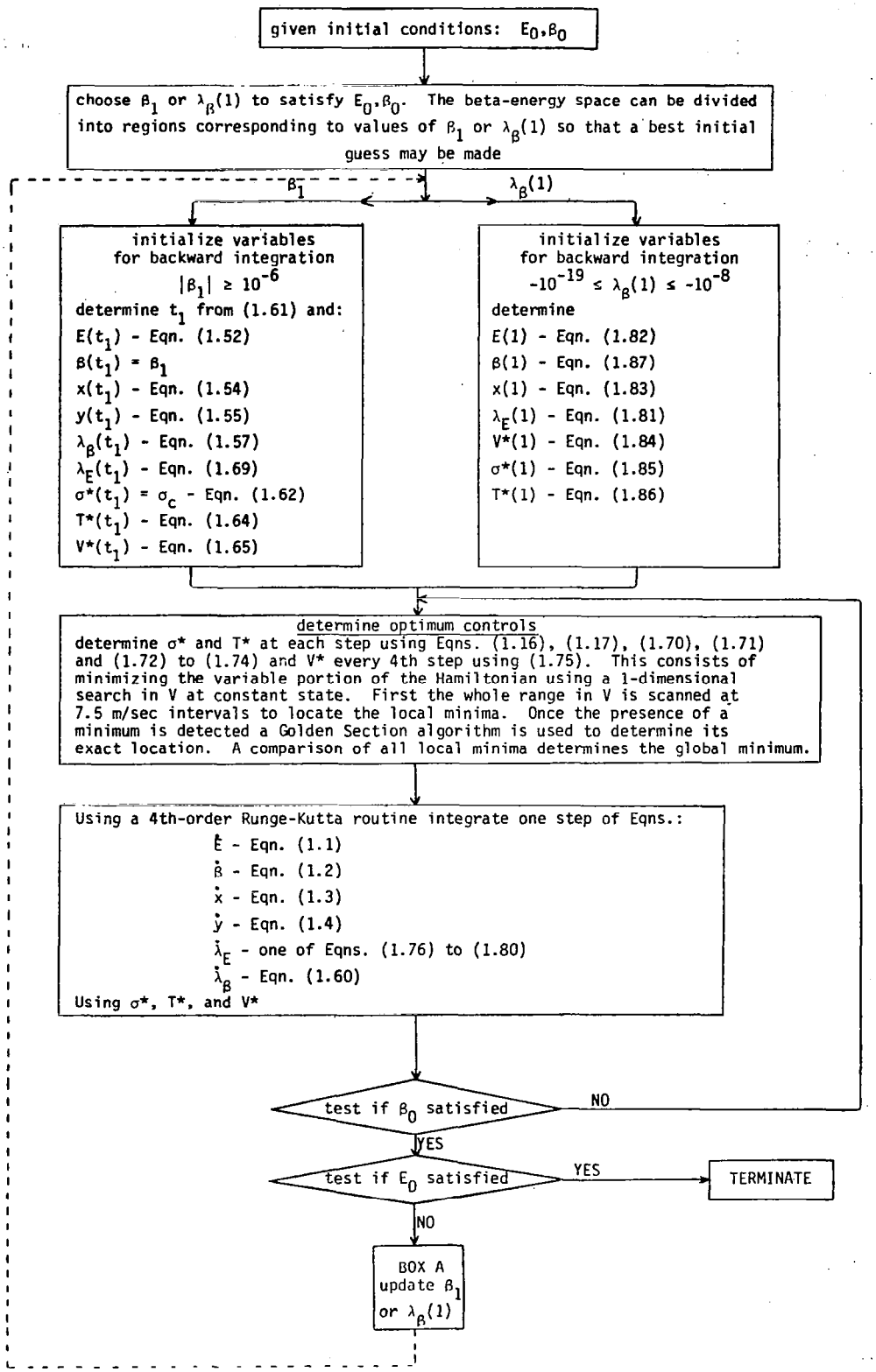


Figure A5.1.6.-Flowchart for Optimal Trajectory.

is an altitude-range profile of the trajectory where the very fast changes in altitude are clearly visible. The variations in Mach number with range appear in Figure A5.1.25. The various combinations of initial conditions (β_0, E_0) that can be reached with this $\lambda_\beta(1)$ value can be seen in Figure A5.1.26 while Figure A5.1.27 shows changes in σ with energy-height. The horizontal plane projection of the initial turn trajectory appears in Figure A5.1.28.

A very interesting view of the trajectory is obtained from the altitude-Mach number profile of Figure A5.1.29. The energy contours have been added in as well as the $|\sigma| = \sigma_m = \sigma_s$ maximum velocity constraint and the $\sigma = 0$ with stall constraint. Starting on the maximum turn locus, the trajectory consists of a dive at approximately constant energy till the velocity has increased to about 0.9 M. The velocity remains approximately constant at 0.9 M while the trajectory gains altitude. Using Figure A5.1.26 which gives the change in β with energy together with the altitude-Mach number profile, the progress of the trajectory can be interpreted with regard to the change in heading angle as well. By the time the heading angle relative to cruise is below about 5°, the trajectory essentially coincides with the $\Delta\beta = 0$ min-time path of Parsons, Bryson and Hoffman (1975). The trajectory follows this path to the V_{max} constraint and then at V_{max} to the cruise point.

A5.1.9 Formulation of the Final Turn

There are two general types of final turns: (1) those beginning with a period of constant energy-maximum thrust turning and (2) those beginning with minimum thrust and straight bank angle chatter. Each of these is now described in detail.

(1) Turns beginning with maximum thrust. The approach here is similar to that for the initial turns. We assume an initial turning segment with $T = T_{max}$, $E = E_c$ and $V = \hat{V}_{max}$ as in Figure A5.1.7.

The minimum time cost function may be expressed as:

$$\min_{t,\sigma,V} \left\{ J = \underbrace{\frac{x_2}{\hat{V}_{max}} - \frac{R_{min}\sin\beta_2}{\hat{V}_{max}}}_{\text{time in cruise}} + \underbrace{\frac{R_{min}\beta_2}{\hat{V}_{max}}}_{\substack{\text{time in turn at} \\ V = \hat{V}_{max}}} + \underbrace{\int_{t_2}^{t_f} dt}_{\substack{\text{time in turn at} \\ V < \hat{V}_{max}}} \right\}$$

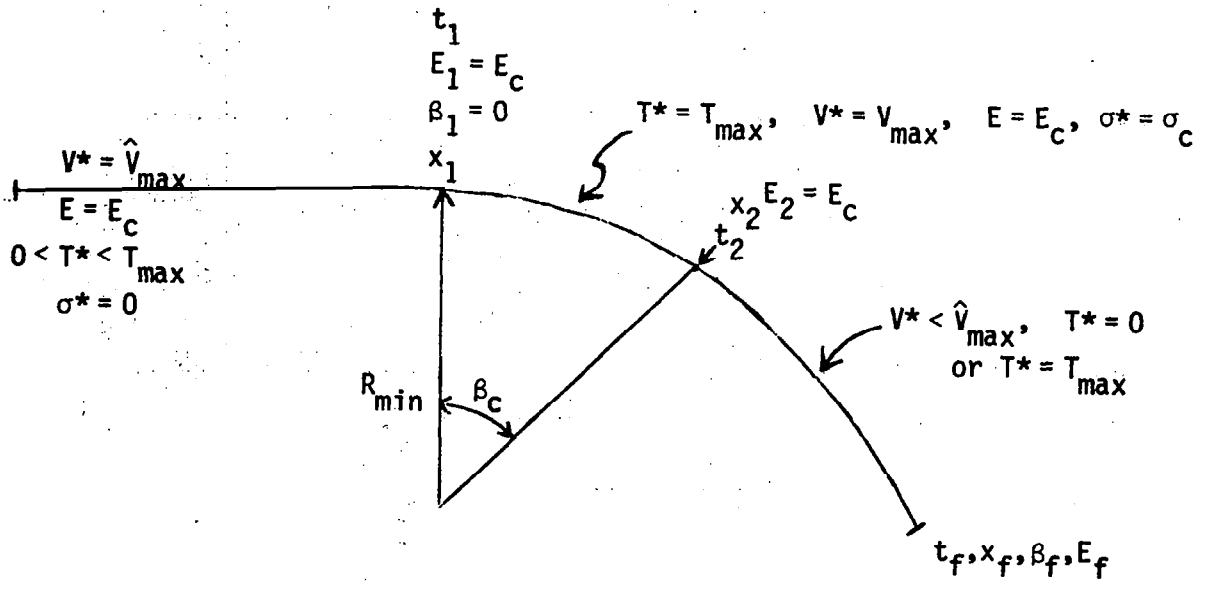


Figure A5.1.7.-Constant Radius Turning Segment Within Final Turn.

subject to equations of motion (1.1) - (1.4), (1.12) - (1.15) and the boundary conditions

$$E(t_2) = E_c \qquad E(t_f) = E_f$$

$$\beta(t_2) = \beta_2, \text{ free parameter} \qquad \beta(t_f) = \beta_f$$

$$x(t_2) = R_{min}\sin\beta_2, \text{ free} \qquad x(t_f) = x_f$$

$$y(t_2) = R_{min}(1 - \cos\beta_2), \text{ free} \qquad y(t_f) = y_f$$

The Hamiltonian for the problem is as in equation (1.27). The first order necessary conditions and first integral are given by (1.29) - (1.37). We now have the following initial adjoints derived as for the initial turn:

$$\lambda_E(t_2) = \frac{-w(1 - \cos\beta_2)}{2D_L\hat{V}_{max}\tan^2\sigma_c}$$

$$\lambda_\beta(t_2) = \frac{-R_{min}(1 - \cos\beta_2)}{\hat{V}_{max}}$$

$$\lambda_x(t_2) = -\frac{1}{\hat{V}_{max}}$$

$$\lambda_y(t_2) = 0$$

The determination of the optimal final turn trajectory closely parallels that of the initial turn as given by equations (1.60) - (1.80), with appropriate subscript changes. The state and adjoint equations are integrated forwards from time t_2 and the problem is parameterized in terms of β_2. At each step the optimum controls are evaluated using equations (1.70) - (1.76).

(2) Turns beginning with minimum thrust and bank angle chatter. These turns require an initial period of straight flight with minimum thrust and bank angle chattering between positive and negative minimum values. This is a maximum deceleration arc and is similar to a singular arc as it only occurs with $\lambda_\beta = 0$. The Hamiltonian with $\lambda_\beta = 0$ is given by:

$$H = 1 + \frac{\lambda_E V}{W}(T - D_0 - D_2\sec^2\sigma) + \lambda_x V\cos\beta + \lambda_y V\sin\beta$$

which is independent of the sign of the bank angle. Application of the Minimum Principle yields:

$$T^* = 0 \quad (\lambda_E > 0)$$

$$\sigma^* = \pm\sigma_{chat}(V)$$

$$V^* = \arg\min\left\{H = 1 - \frac{\lambda_E V}{W}(D_0 + D_L\sec^2\sigma_{chat}) + \lambda_x V\cos\beta_s + \lambda_y V\sin\beta_s\right\}$$

where $\sigma_{chat}(V) = \min[\sigma_s(V), \sigma_m]$

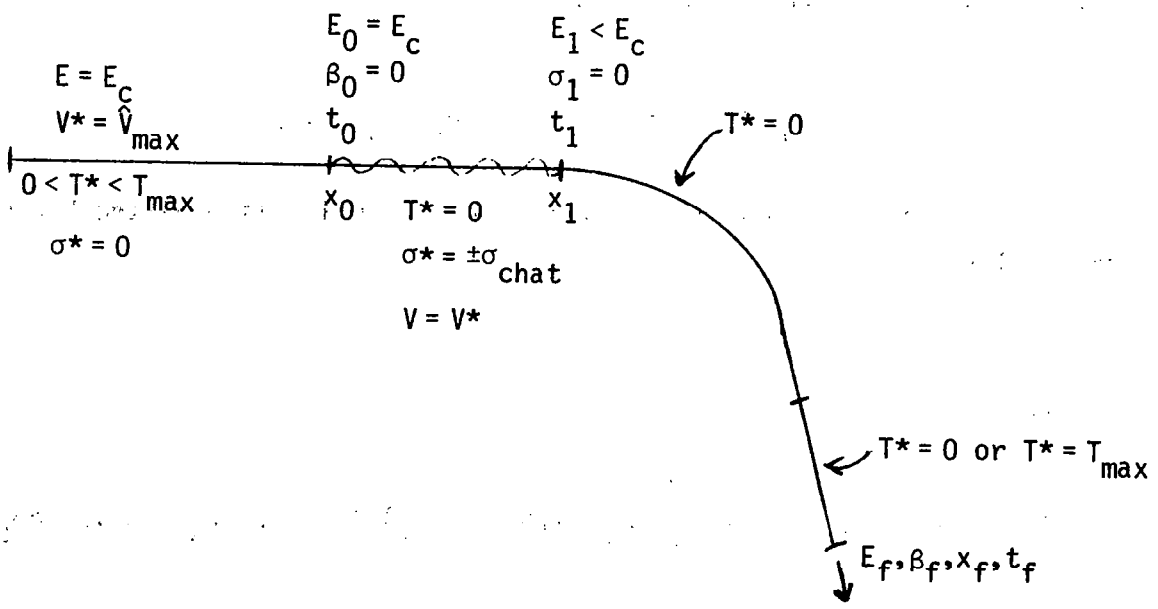
β_s = cruise heading

Note that as a result of our modeling, bank angle can be changed instantaneously and so an equal time chatter between positive and negative bank angle can be made so that $\beta = 0$ on the average and hence λ_β remains zero. For a better understanding of the geometry of the problem consider the Minimum Principle applied to the problem with $\beta_s = 0$ so that $\lambda_y = 0$:

$$\min_{T,\sigma,V} = (\lambda_\beta, \lambda_E, \lambda_x) \cdot (\dot{\beta}, \dot{E}, \dot{x})^T$$

For a given state the state-velocity space (hodograph) appears approximately as in Figure A5.1.8 considering just a single section in each plane. This approach was an aid in the determination of optimal velocity when $T^* = 0$. As in Figure A5.1.8, the admissible state-velocity space has a non-convex nature so that after leaving the cruise arc at $V^* = V_{max}$, the solution can be expected to jump at some point along the trajectory from the $T^* = 0$, $V^* = V_{max}$, $\sigma^* = \sigma_m$ corner to the $T^* = 0$, V^* for $|\sigma| = \sigma_s = \sigma_m$, $|\sigma^*| = \sigma_s = \sigma_m$ corner of the state-velocity space. This knowledge reduced the one-dimensional search in velocity described in equation (1.75) to just a check at these two velocities. As a result computation time was greatly reduced.

The turns with bank angle chatter may be formulated as follows:



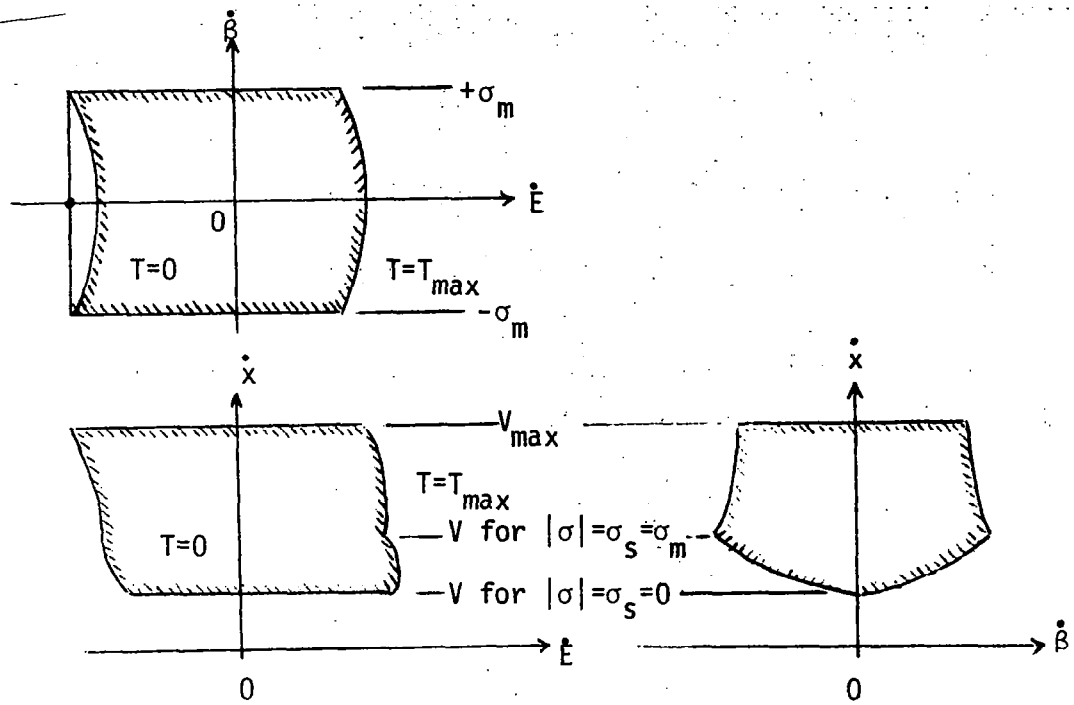


Figure A5.1.8.-Sections Through State-Velocity Space (Hodograph).

with the cost function expressed as:

$$\min_{T,V,\sigma} \left\{ J = \frac{x_0}{\hat{V}_{max}} + \int_{E_c}^{E_1} \frac{dE}{\dot{E}} \Bigg|_{\substack{V = V^* \\ T^* = 0 \\ \sigma^* = \pm\sigma_{chat}}} + \int_{t_1}^{t_f} dt \right\}$$

time in cruise | time in chatter | time in turn

which is

$$\min_{T,V,\sigma} \left\{ J = \frac{x_1}{\hat{V}_{max}} + \int_{E_c}^{E_1} \frac{\left(1 - \frac{V}{\hat{V}_{max}}\right)}{\dot{E}} dE \Bigg|_{\substack{T^* = 0 \\ \sigma^* = \pm\sigma_{chat} \\ V = V^*}} + \int_{t_1}^{t_f} dt \right\}$$

subject to equations (1.1) - (1.4) and (1.12) - (1.15) and the boundary conditions

$$E(t_1) = E_1, \text{ free parameter} \qquad E(t_f) = E_f$$

$$\beta(t_1) = 0 \qquad \beta(t_f) = \beta_f$$

$$x(t_1) = x_1(E_1), \text{ free} \qquad x(t_f) = x_f$$

$$y(t_1) = \text{free} \qquad y(t_f) = y_f$$

The Hamiltonian and first-order conditions are the same as before: (1.27) and (1.29) - (1.37) with the initial adjoints

$$\lambda_E(t_1) = -\left.\frac{\left(1 - \frac{V}{\hat{V}_{max}}\right)}{\dot{E}}\right|_{\substack{T^* = 0 \\ \sigma^* = \pm\sigma_{chat} \\ V = V^*}}$$

$\lambda_\beta(t_1) = 0$ for continuity across juncture with chatter arc

$$\lambda_x(t_1) = \frac{-1}{\hat{V}_{max}}$$

$$\lambda_y(t_1) = 0$$

Again we have the problem parameterized in terms of a single parameter E_1.

A5.1.10 Numerical Results

Parsons (1972) presents numerical results using the theory described above. Figure A5.1.9 gives the family of trajectories in the beta-energy plane and the regions marked I to VIII are described by means of the control programs in Figure A5.1.10. For the chatter cases t_1 has to be determined by integration of the equation:

$$t_1 = \left.\int_{E_c}^{E_1} \frac{dE}{\dot{E}}\right|_{\substack{V = V^* \\ T^* = 0 \\ \sigma^* = \pm\sigma_{chat}}}$$

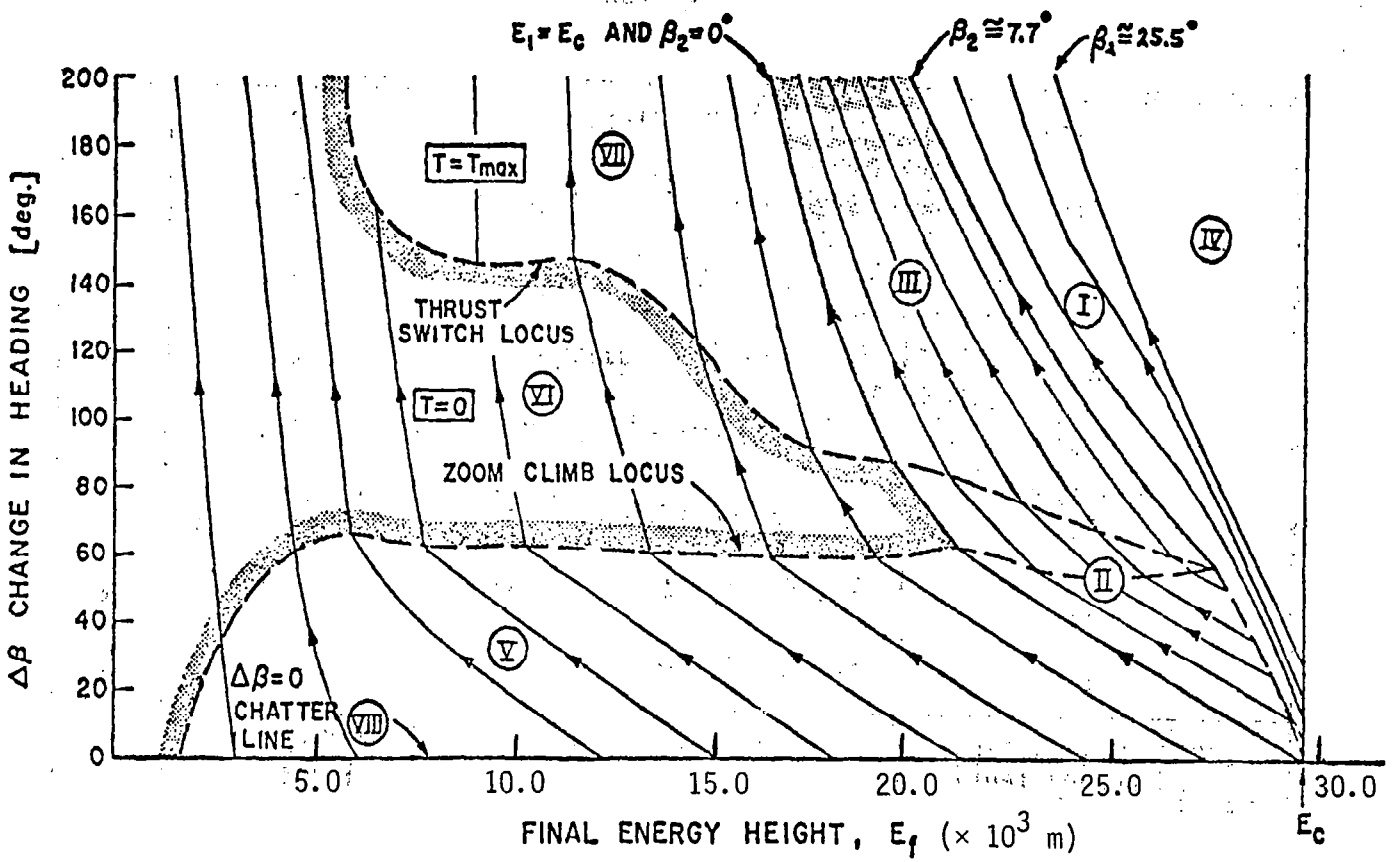


Figure A5.1.9.-Change in Heading for Variable-Altitude Final Turns From Cruise Arc.

Type	Thrust Program	Bank Angle Program	Final Region or Line of Fig. 3.5.3
1	T_{max}	σ_c, $\sigma_c < \sigma < \sigma_m$, σ_m, $\sigma_m = \sigma_s$	I
2	T_{max}, 0	σ_c, $\sigma_c < \sigma < \sigma_m$, σ_m, $\sigma_m = \sigma_s$	II
3	T_{max}, 0, T_{max}	σ_c, $\sigma_c < \sigma < \sigma_m$, σ_m, $\sigma_m = \sigma_s$	III
4	T_{max}	$\sigma_c \neq \sigma < \sigma_m$, σ_m, $\sigma_m = \sigma_s$	IV
5	0	chatter $\pm\sigma_{chat}$, σ_m	V
6	0	chatter $\pm\sigma_{chat}$, σ_m, $\sigma_m = \sigma_s$	VI
7	0, T_{max}	chatter $\pm\sigma_{chat}$, σ_m, $\sigma_m = \sigma_s$	VII
8	0	chatter $\pm\sigma_{chat}$	VIII

Figure A5.1.10

Summary of Control Programs for Variable-Altitude Final Turns from Cruise Arc

and then the four states and two adjoints were integrated from t_1 to t_f. As in Figure A5.1.9 the zoom climb when V* jumps is followed after a short while on the max-turn locus by a switch to T_{max} from T = 0. As explained earlier, computation time here was considerably lower than that in the initial turns because of the knowledge of the non-convex nature of the admissible state-velocity space reducing the search for V* to just two points.

Trajectories beginning with maximum thrust were of the same form as those with chatter except that there is no straight flight portion. In the altitude-Mach space, the trajectories move down the maximum velocity constraint and then zoom to the max-turn locus as for the example case. The constant energy turning is with $T = T_{max}$ followed by T = 0 on the max-velocity constraint through the zoom and onto the locus. After a short stay on the locus, thrust switches back to max. These turns are suitable when the final energy is high, that is, around 24 - 28 Km.

A5.1.11 Computational Efficiency

In obtaining the example trajectory of Figures A5.1.11 to A5.1.29, a ratio of about 2:1 resulted for CPU time:real time. It is possible to reduce this ratio to at least 1:2 by making the program more efficient. It is worth mentioning that substantial improvements were made to arrive at the value 2:1 as initially the ratio was of the order of 10:1. This was achieved by allowing a tolerance of 15.2 m and 0.001 M on the spline fits for all the tabulated data, i.e., for changes in altitude and Mach number less than those above, new values of aerodynamic or atmospheric data were not obtained. The sacrifice in accuracy was minimal. Another time-saving method was to increase the storage in the computer (e.g., spline coefficients) so as to avoid calling standard spline fit subroutines too many times.

In analyzing the distribution of CPU time among the major computations required, it was found that the calculation of V* was heavy on time. Hence, the frequency of this computation was reduced to every $4^{\underline{th}}$ step which resulted in a considerable decrease in CPU time. The sacrifice in accuracy began to show up when the frequency was reduced to less that every $4^{\underline{th}}$ step. In order to understand the structural form of the Hamiltonian F(v), the curve was plotted out at various stages along the trajectory. Figures A5.1.30 - A5.1.42 cover the major

changes in F(V) and hence V*. As expected, these occur along the zoom maneuvers where velocity is traded for altitude at constant energy and in very little time. An anticipation of when these jumps in V* occur (i.e., when zoom maneuvers are made) will allow a reduction in the range of V over which the search is made for V* at points along the trajectory between zoom dives.

For real-time implementation (Subsection 5.5.2) computation time is considerably reduced by storing parts of the trajectory (e.g., the $\Delta\beta = 0$ minimum time path).

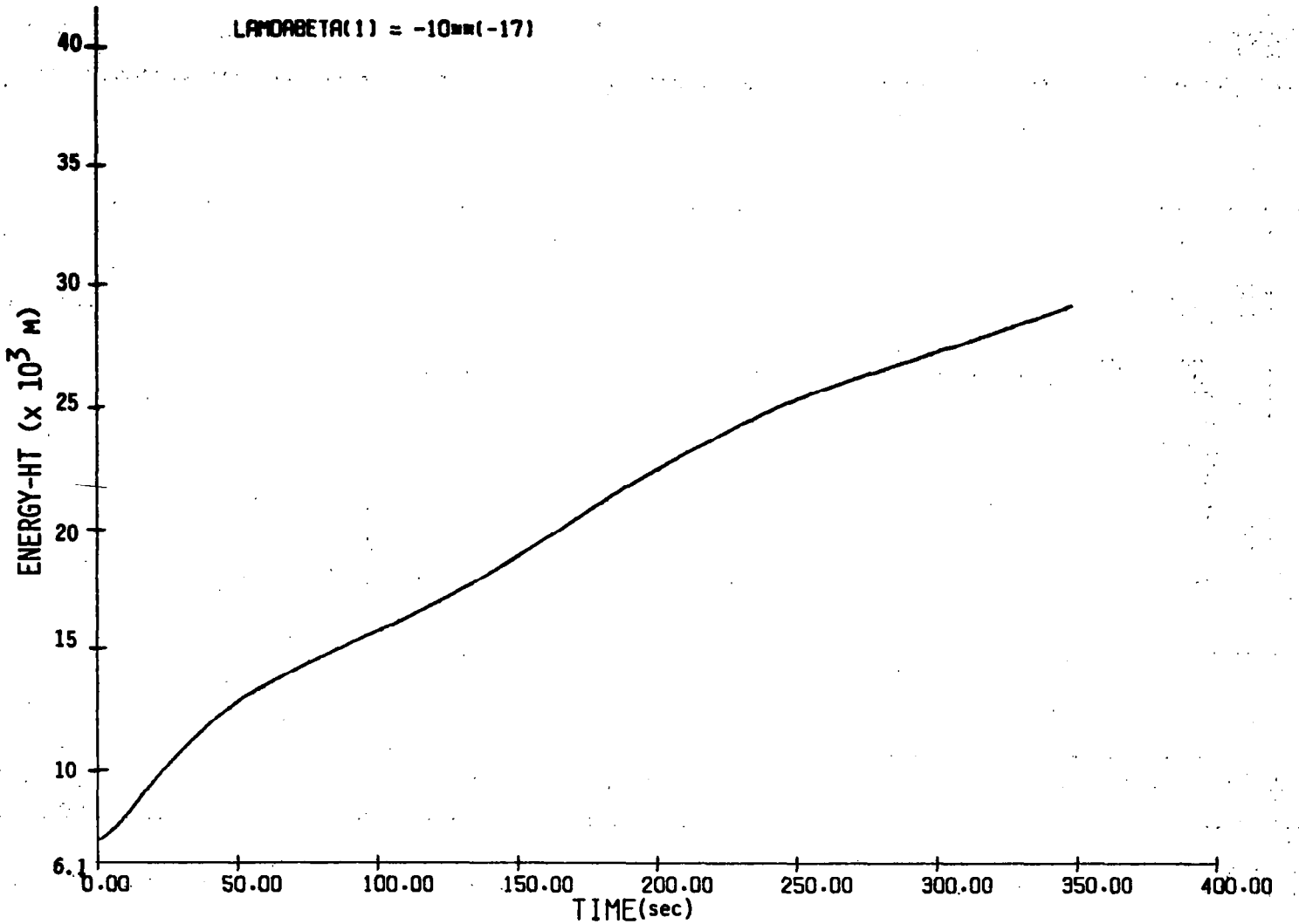


Figure A5.1.11 Energy-ht. time history

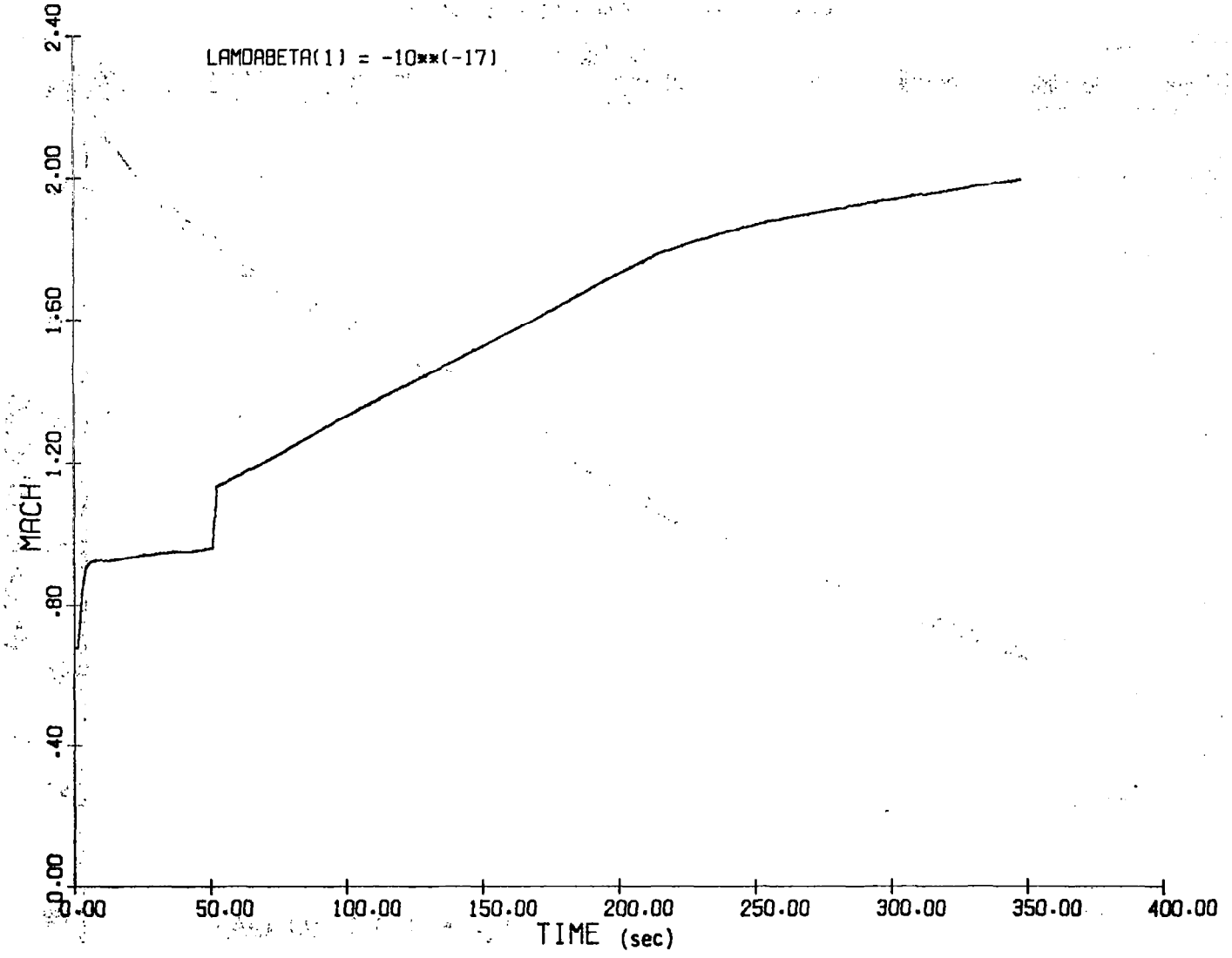


Figure A5.1.12 Mach number variations with time

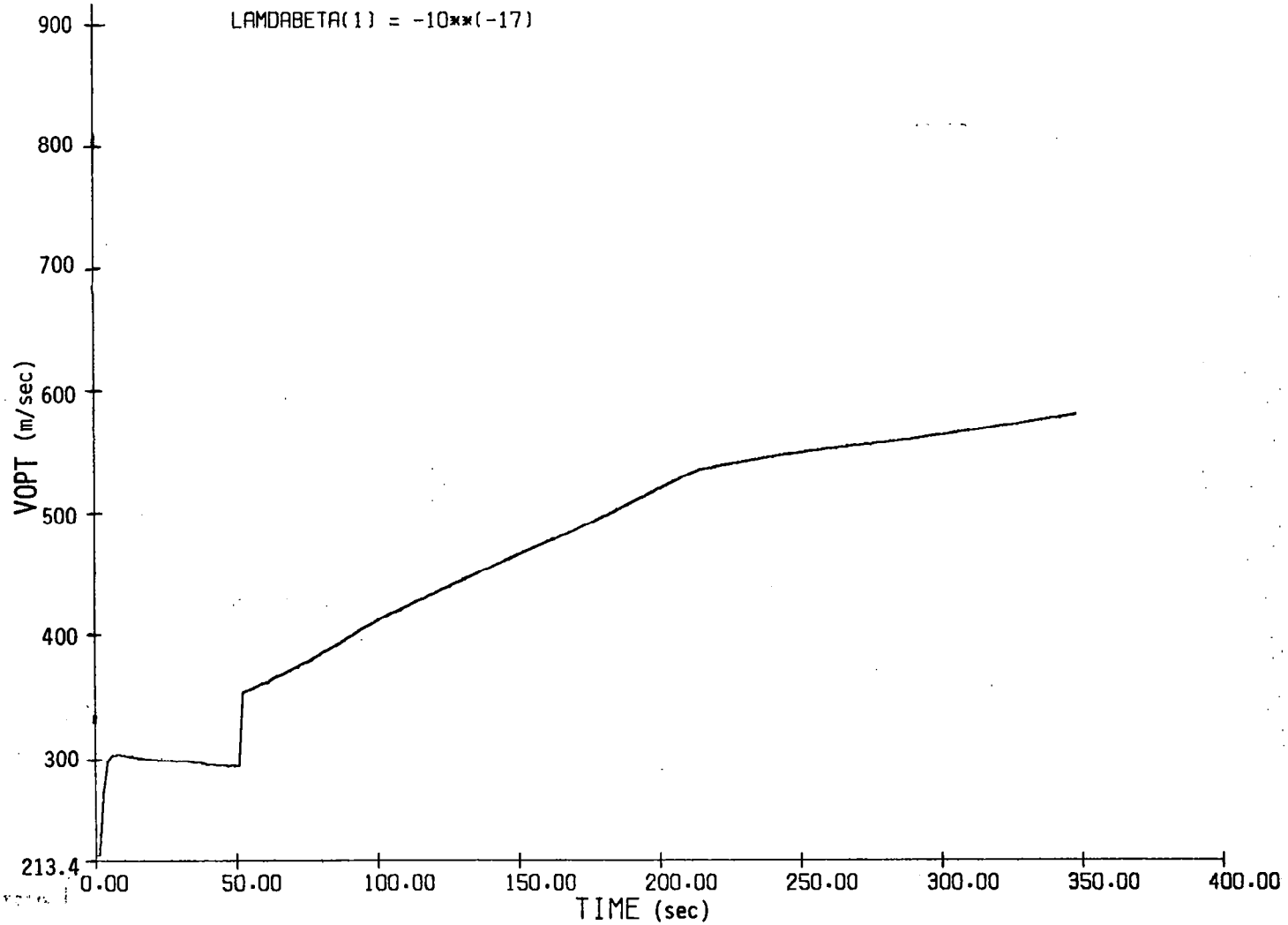


Figure A5.1.13 Changes in velocity with time

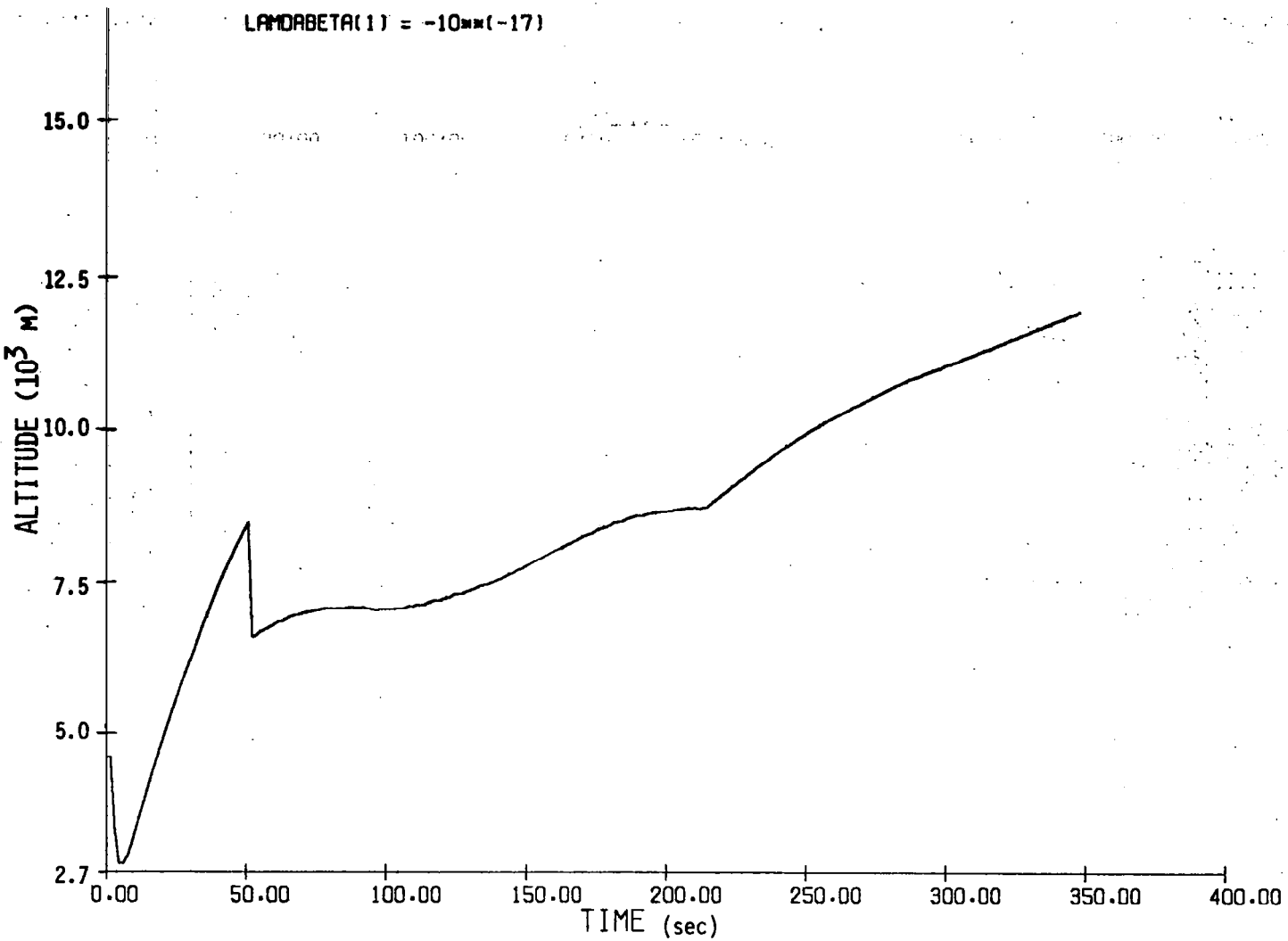


Figure A5.1.14 Changes in altitude with time

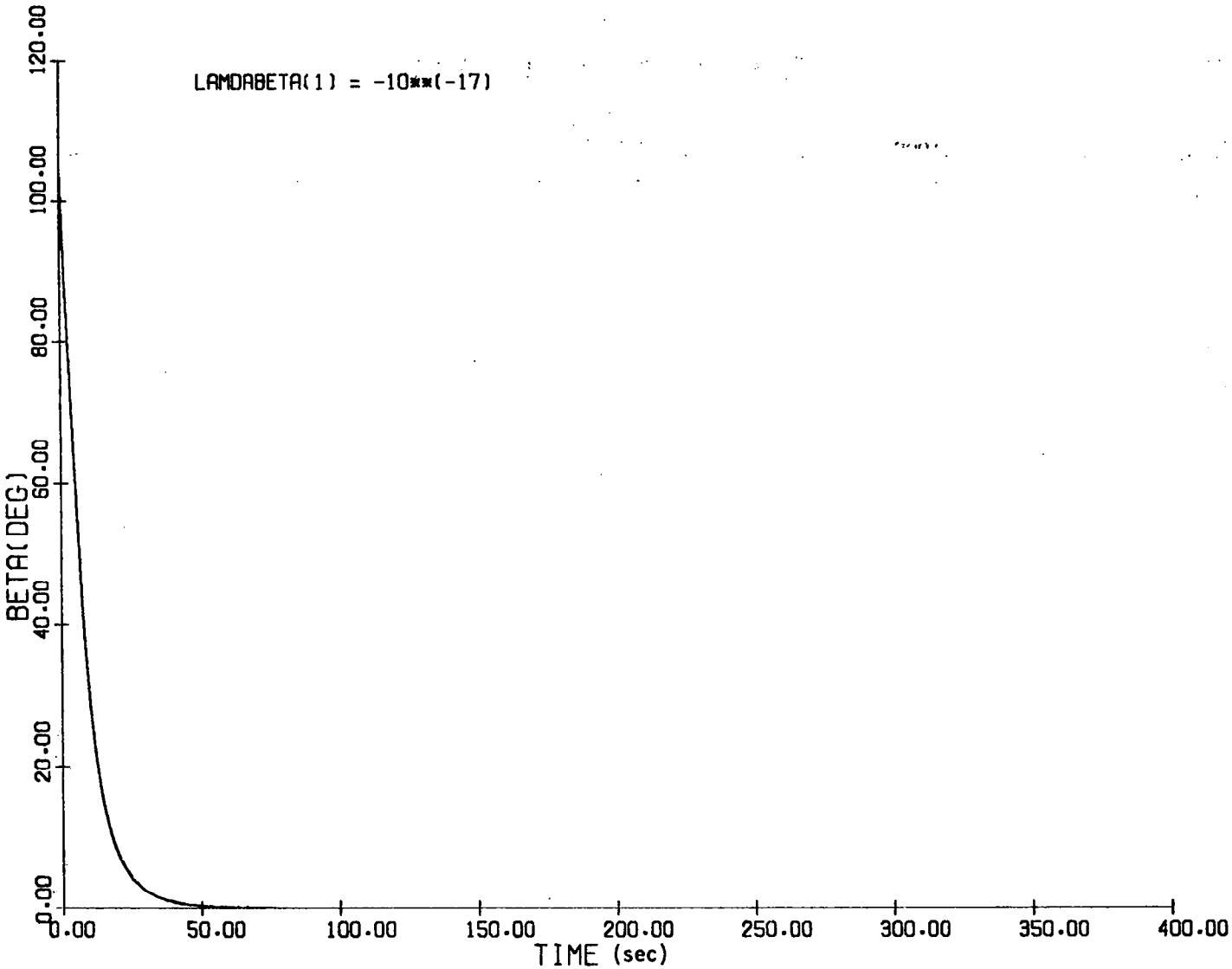


Figure A5.1.15 Change in heading with time

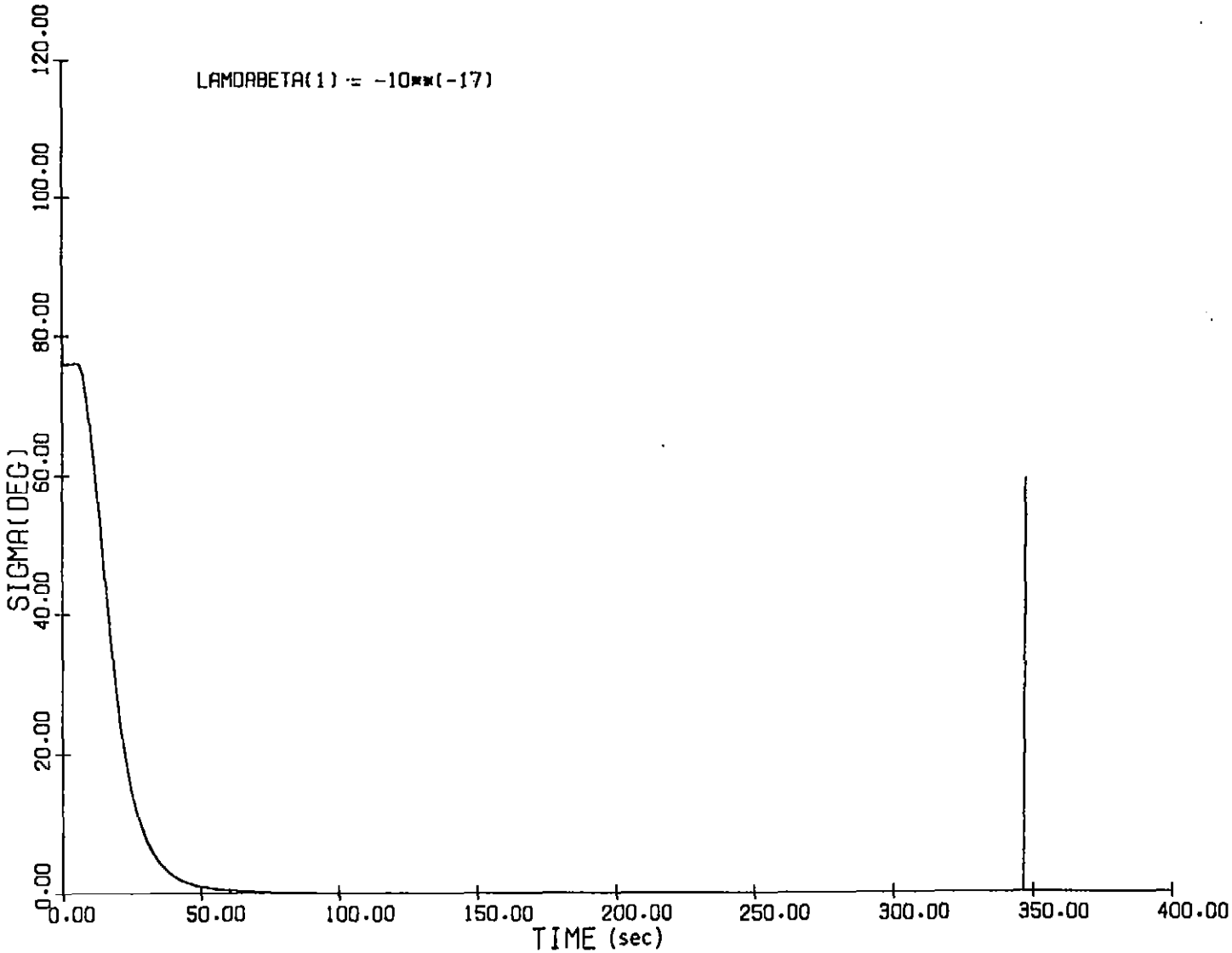


Figure A5.1.16 Bank angle time history

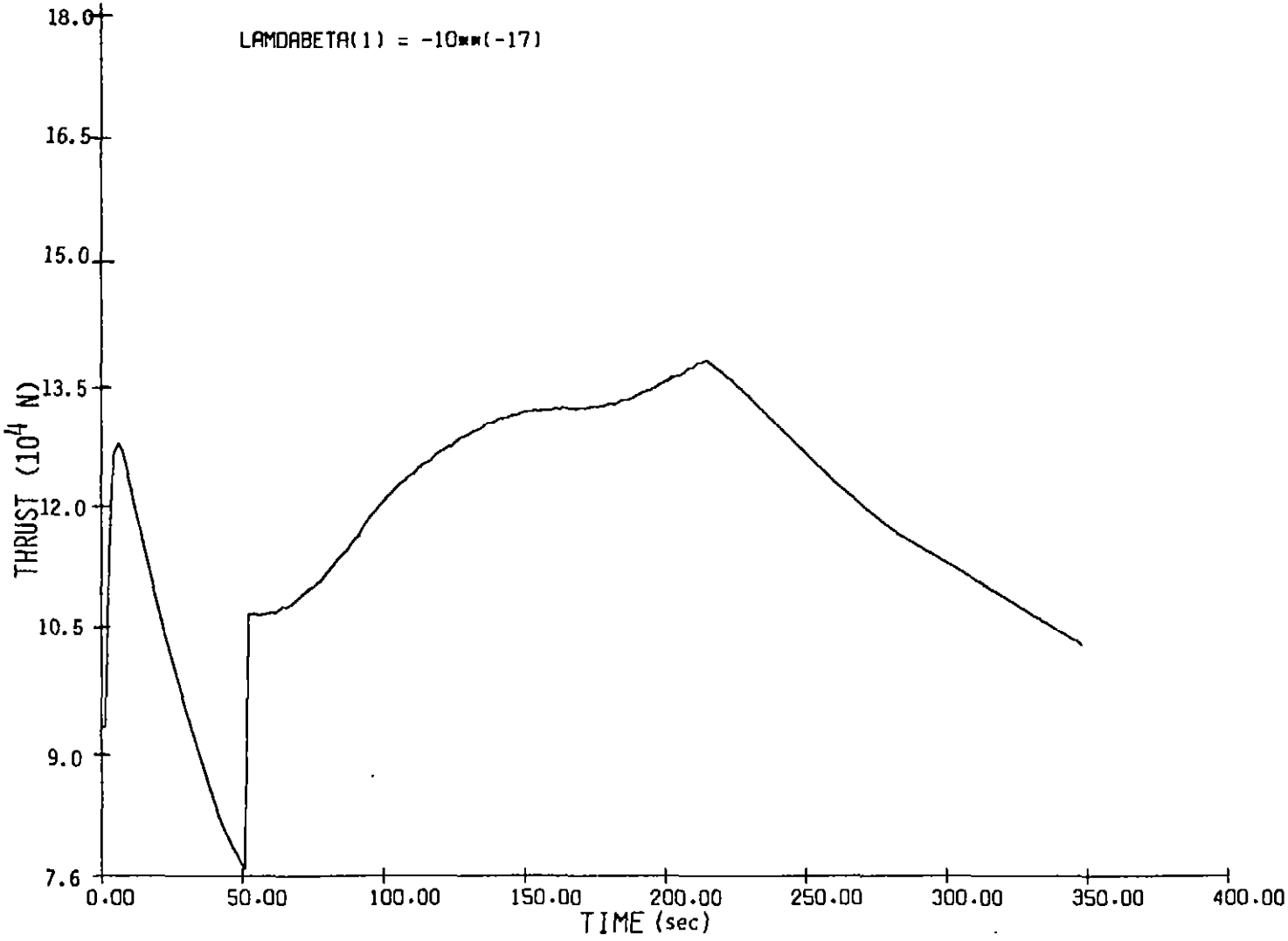


Figure A5.1.17 Variations in thrust with time

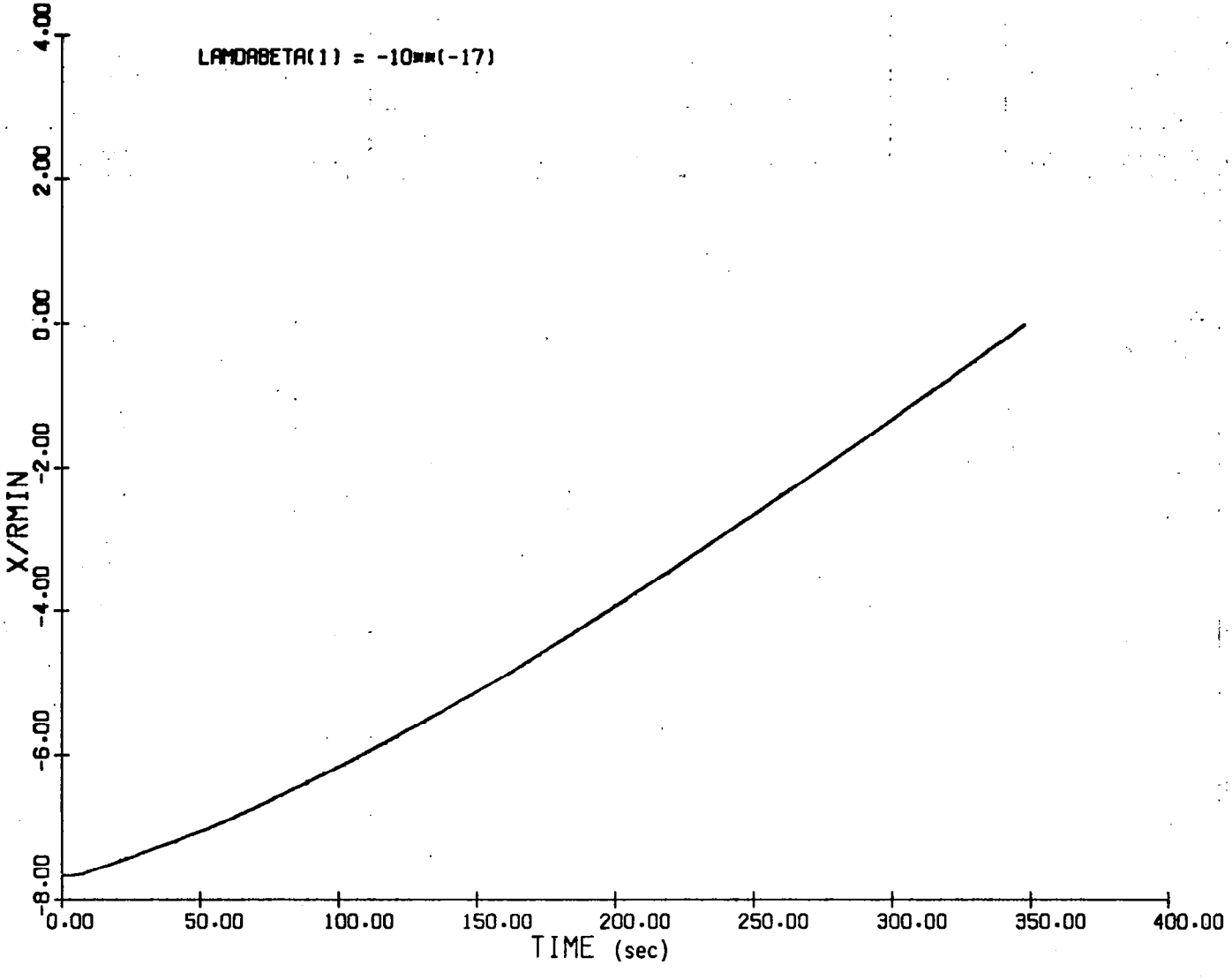


Figure A5.1.18 x-component of horizontal plane projection of trajectory (x=0 when cruise point is reached)

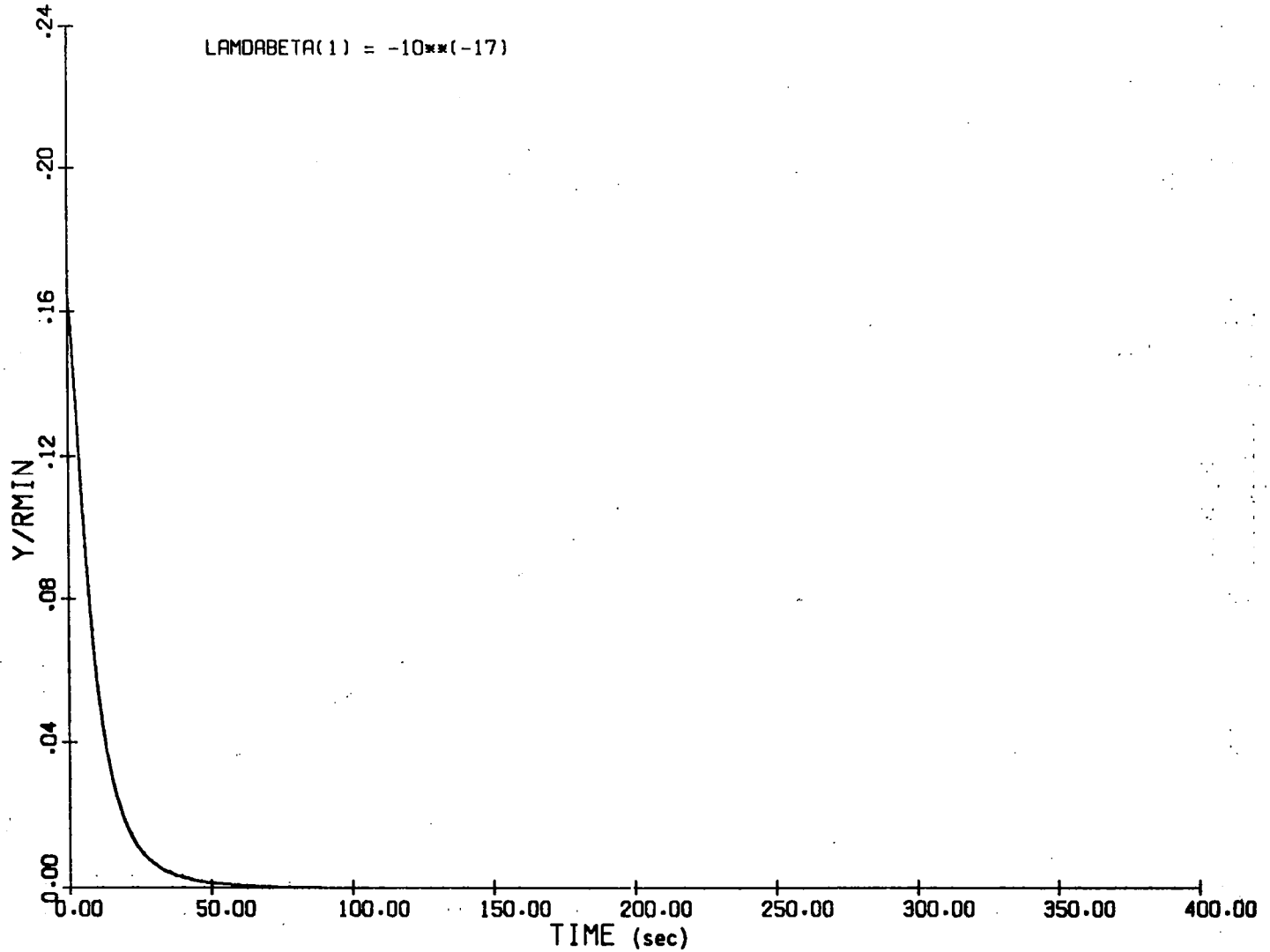


Figure A5.1.19 y-component of horizontal plane projection of trajectory (y=0 when cruise point is reached)

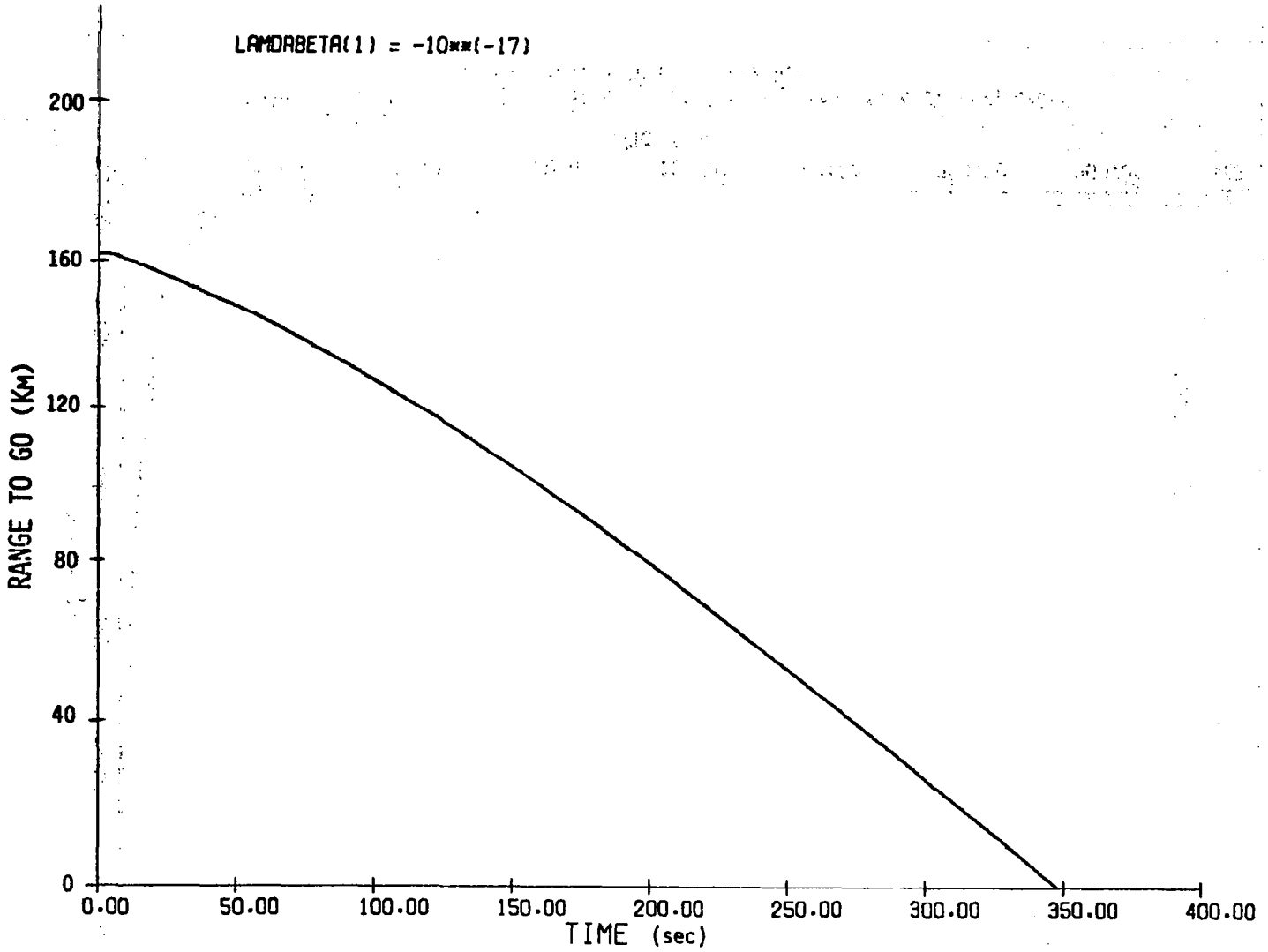


Figure A5.1.20 Range-to-go in horizontal plane with time

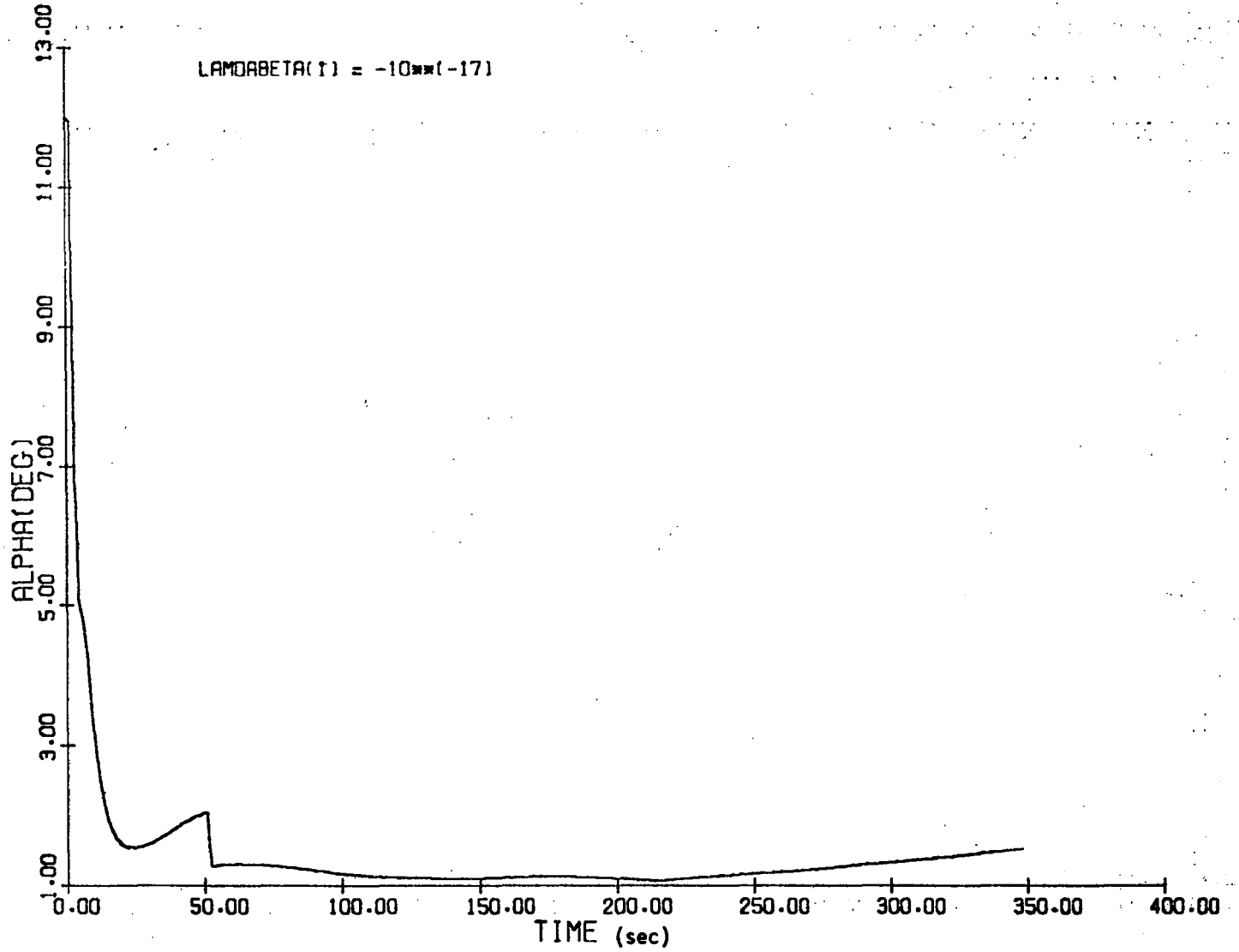


Figure A5.1.21 Changes in alpha with time

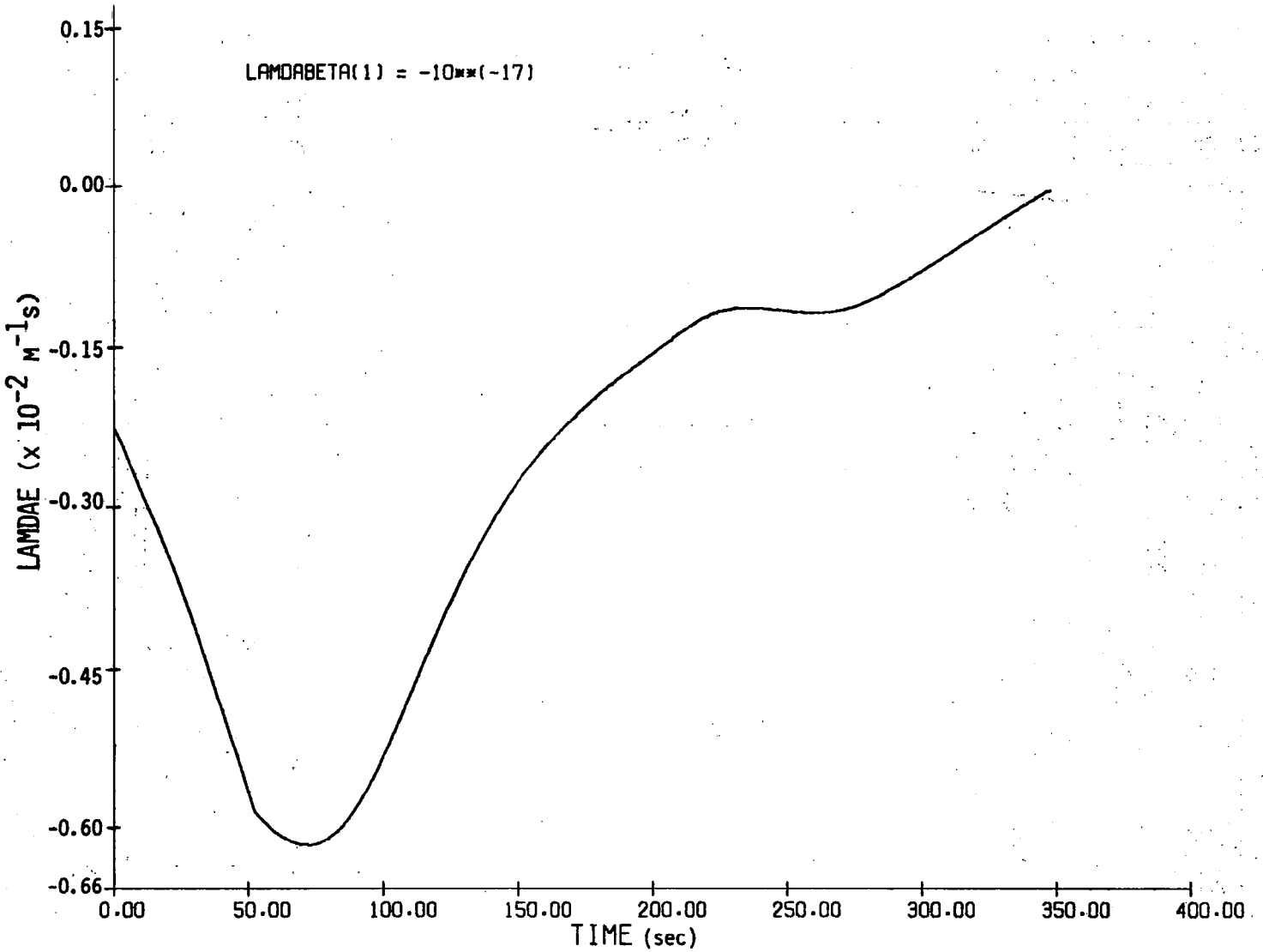


Figure A5.1.22 λ_E time-history

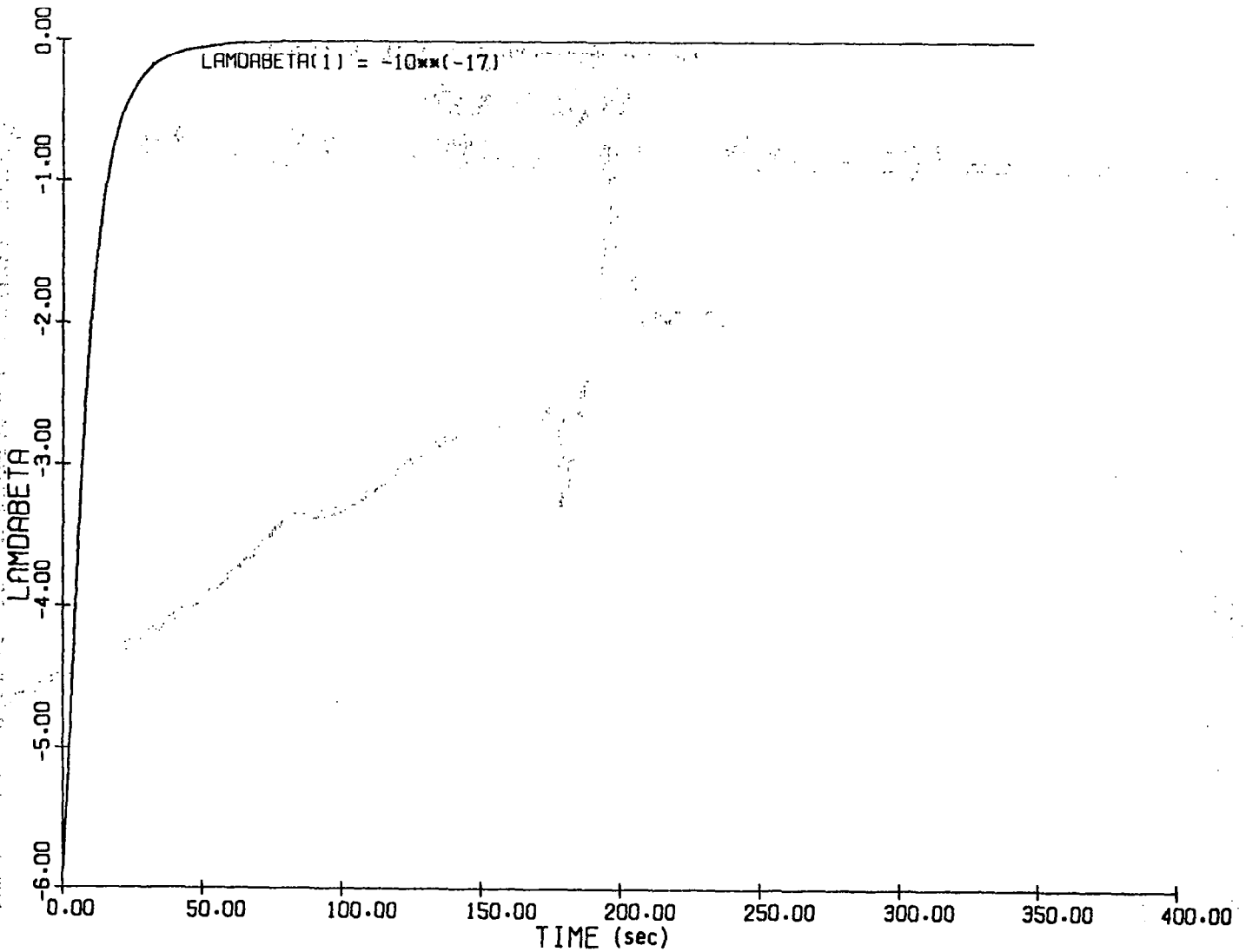


Figure A5.1.23 λ_β time-history

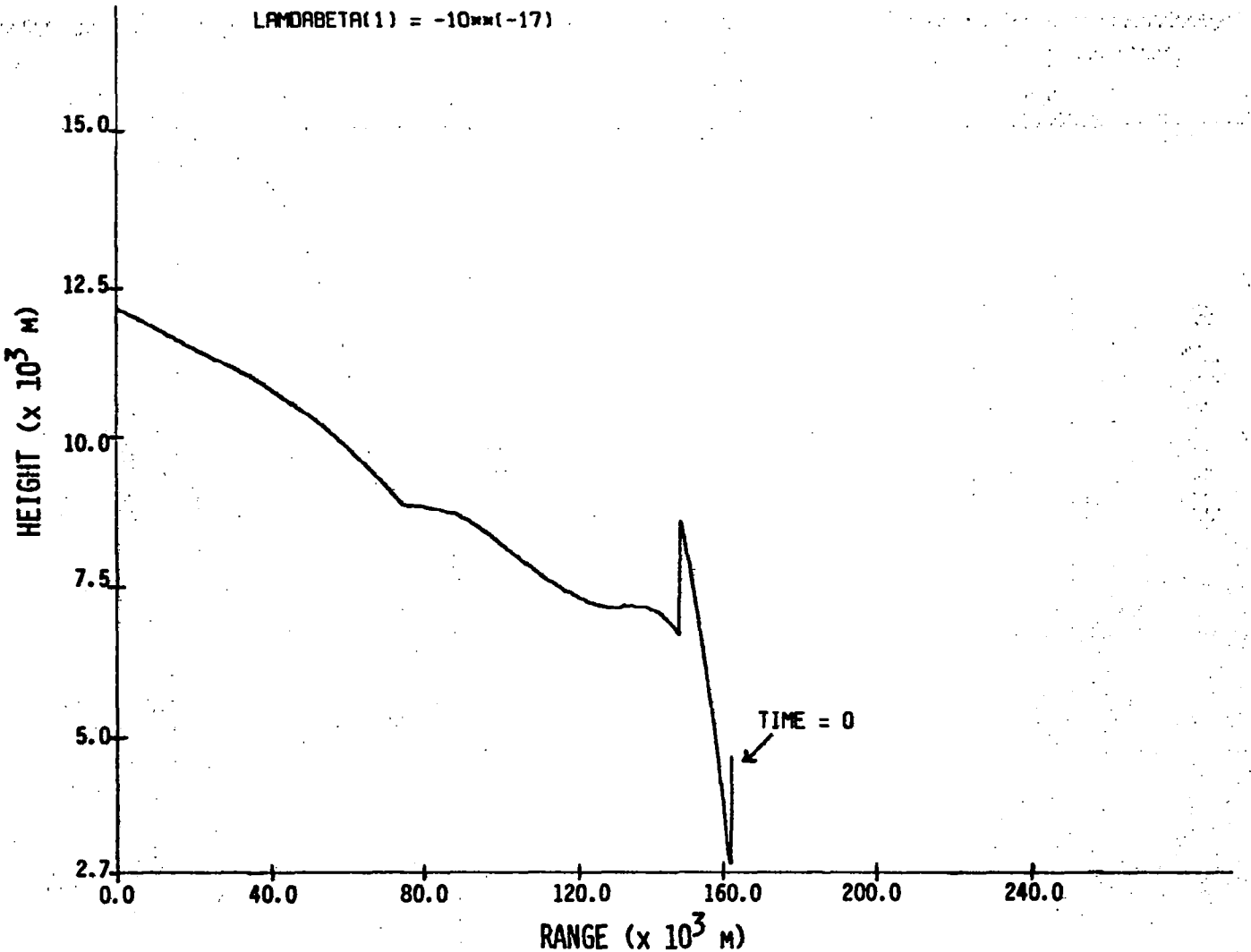


Figure A5.1.24 Altitude-range profile

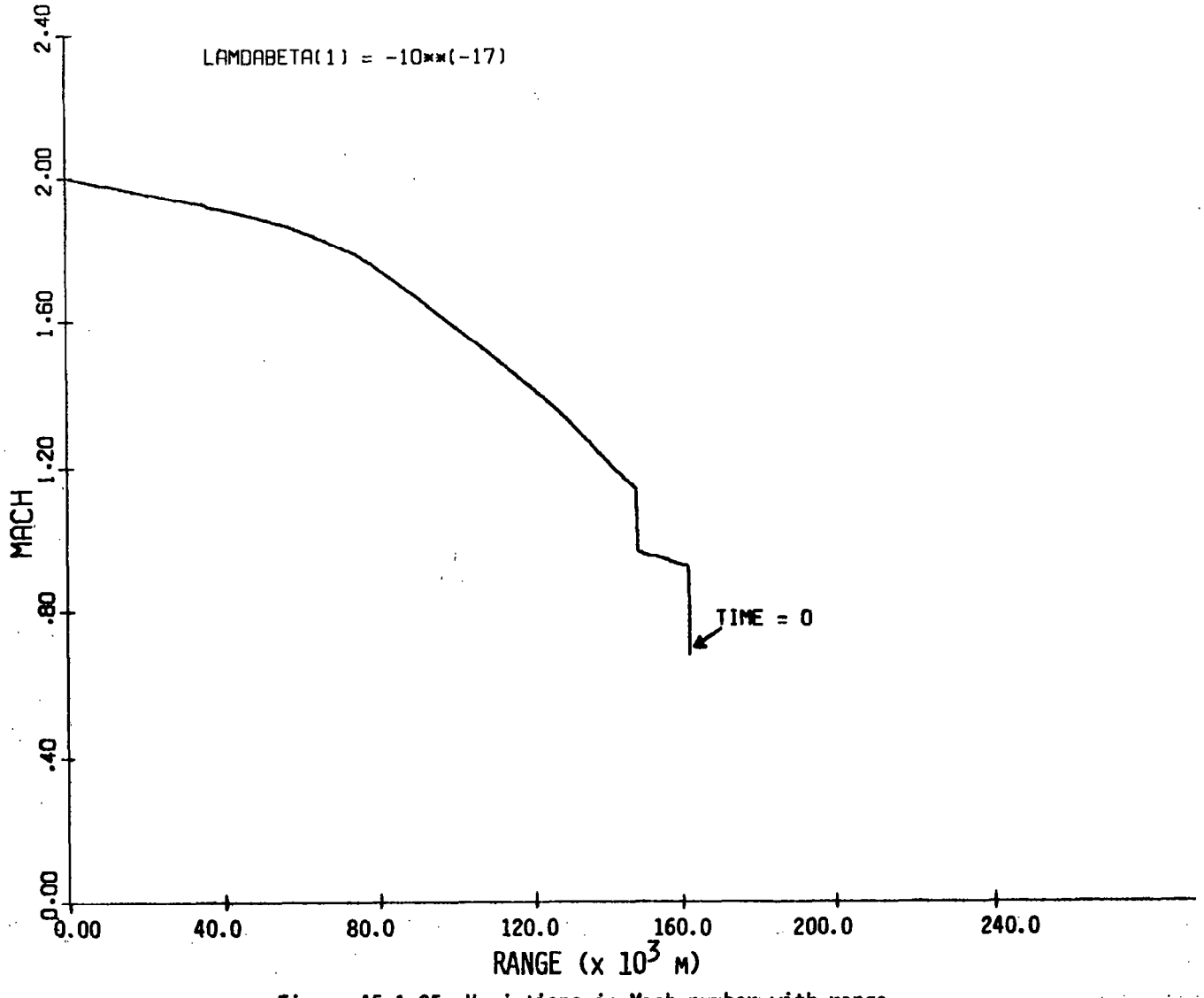


Figure A5.1.25 Variations in Mach number with range

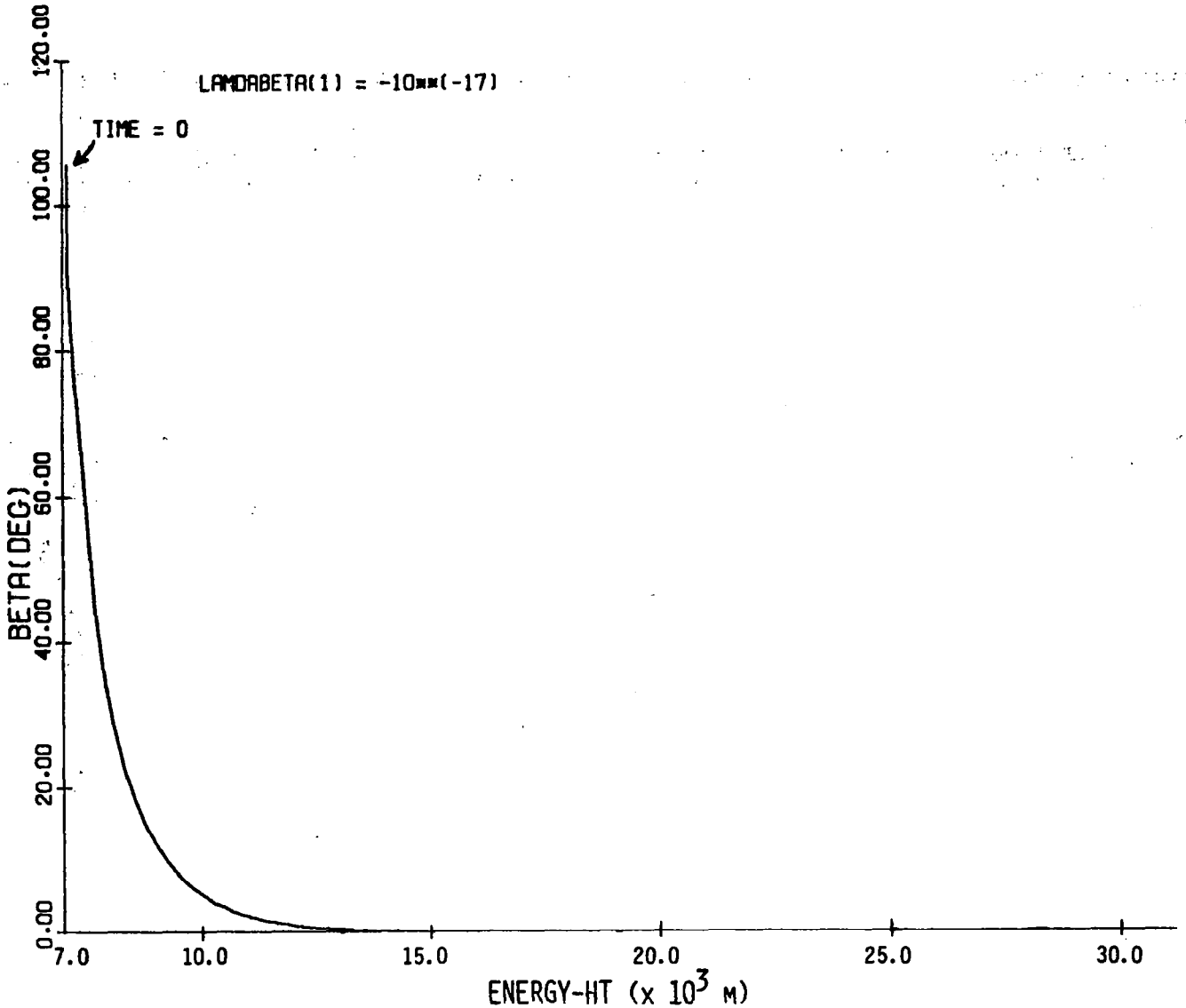


Figure A5.1.26 Change in heading with energy-ht

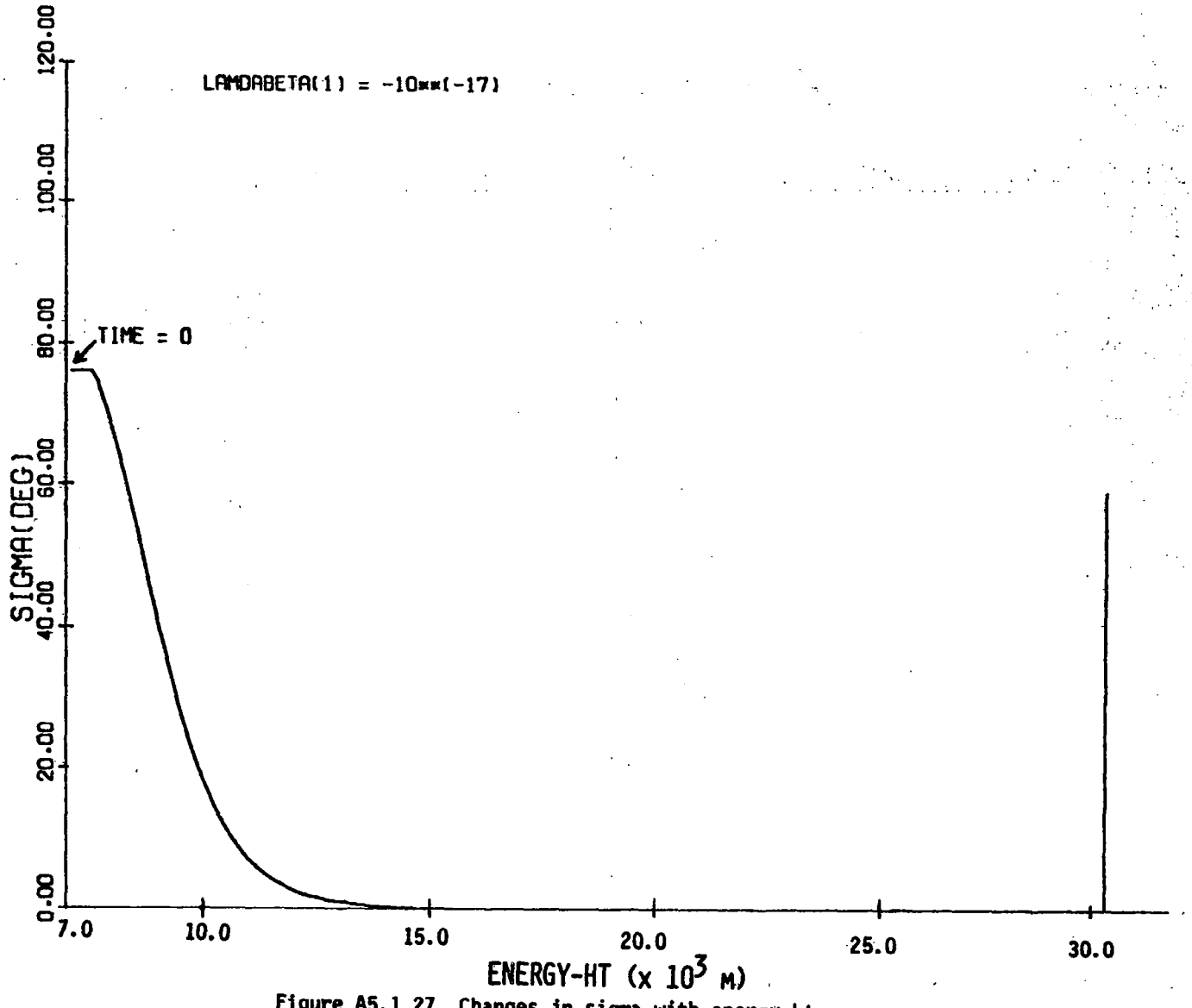


Figure A5.1.27 Changes in sigma with energy-ht

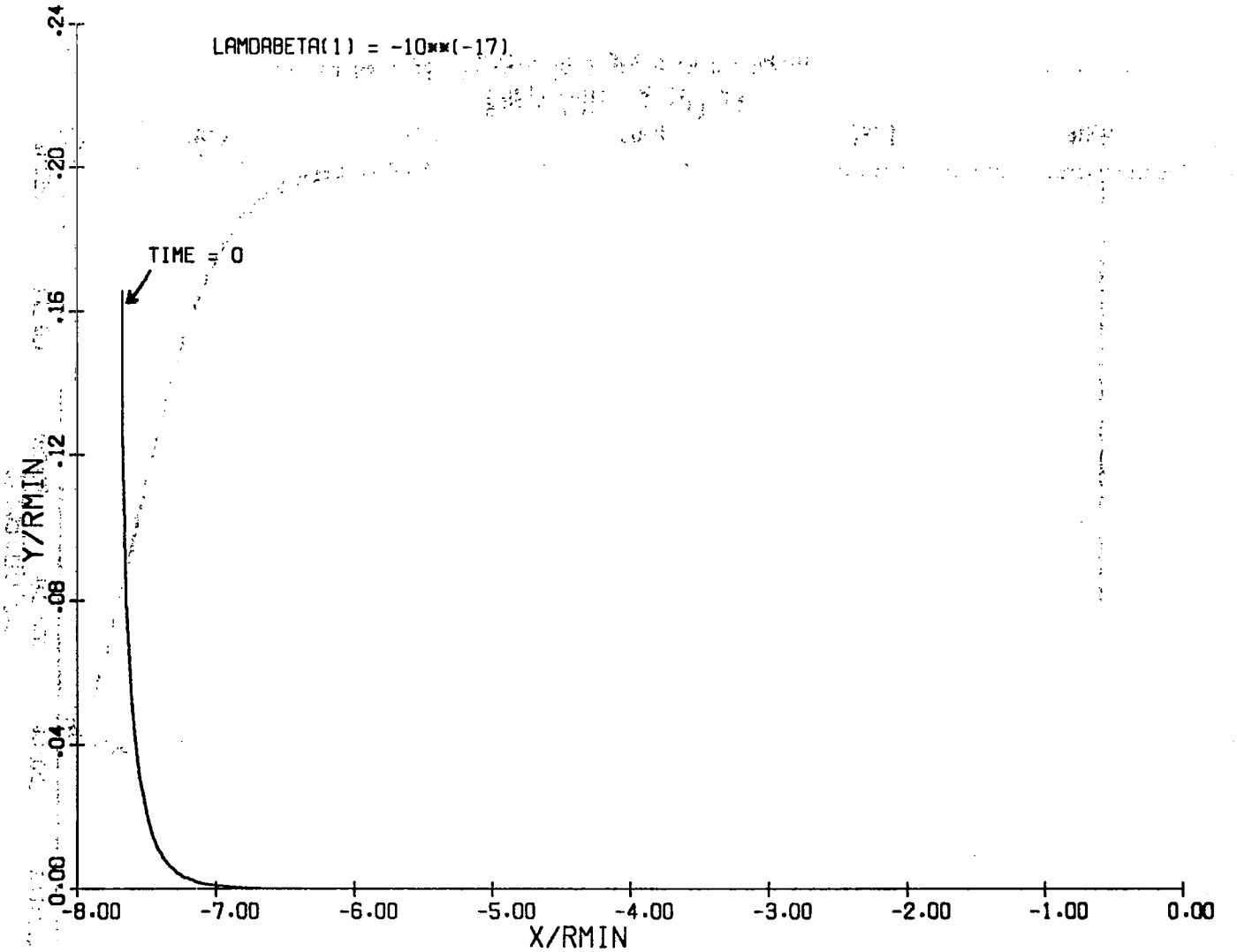


Figure A5.1.28 Horizontal plane projection of trajectory (x=0, y=0 when cruise point is reached)

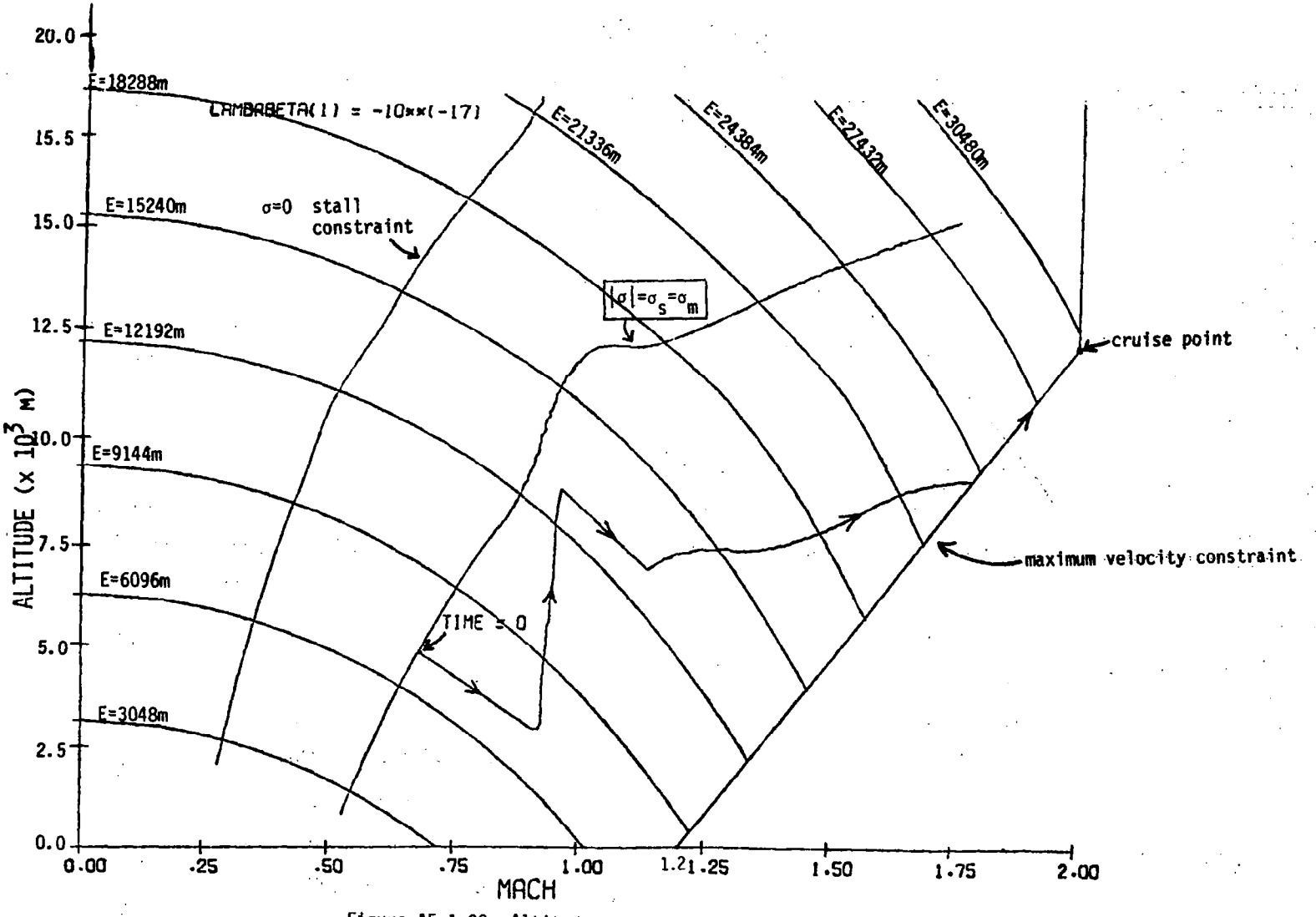


Figure A5.1.29 Altitude-Mach number profile

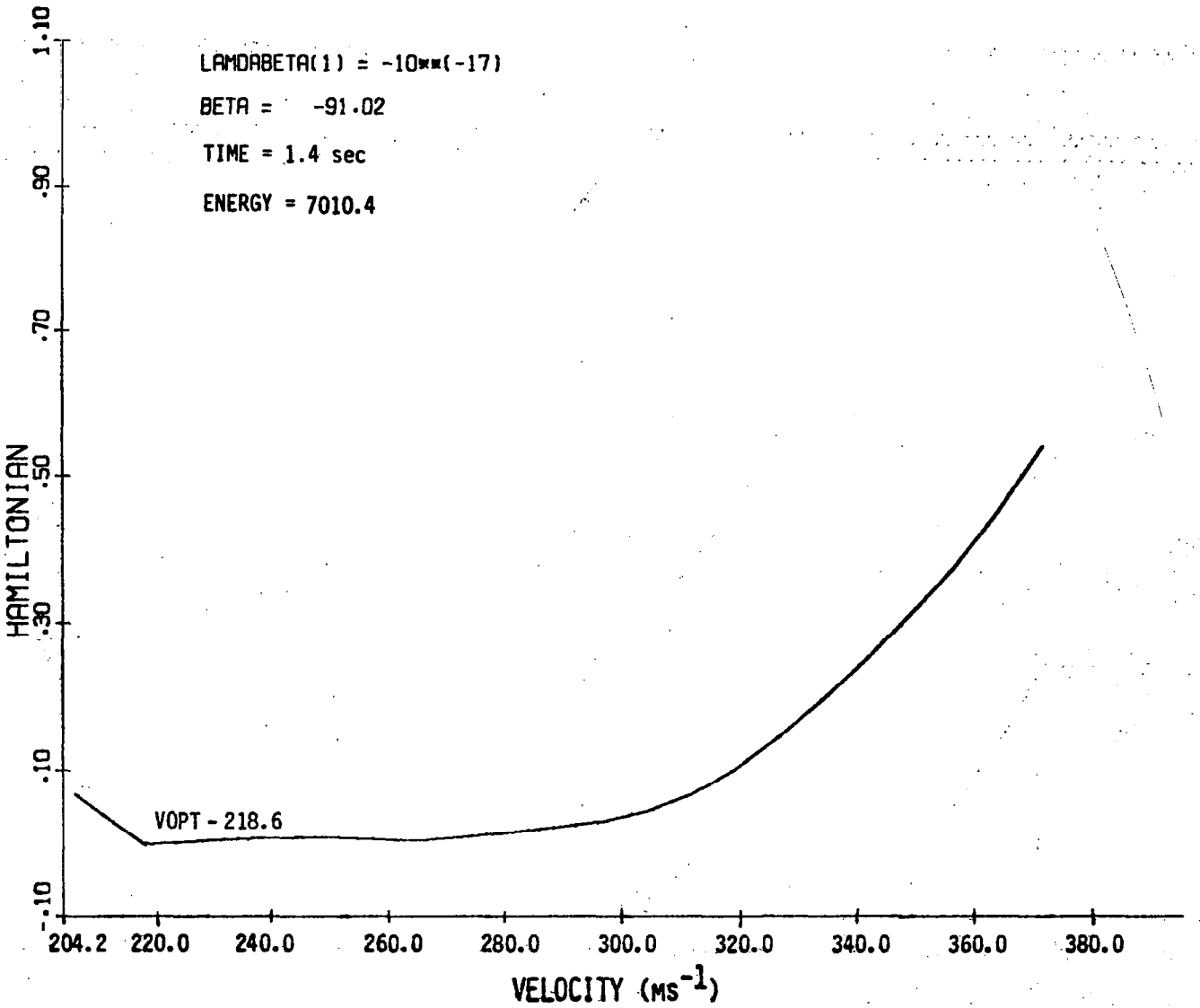


Figure A5.1.30 F(V) at the beginning of the first zoom dive.

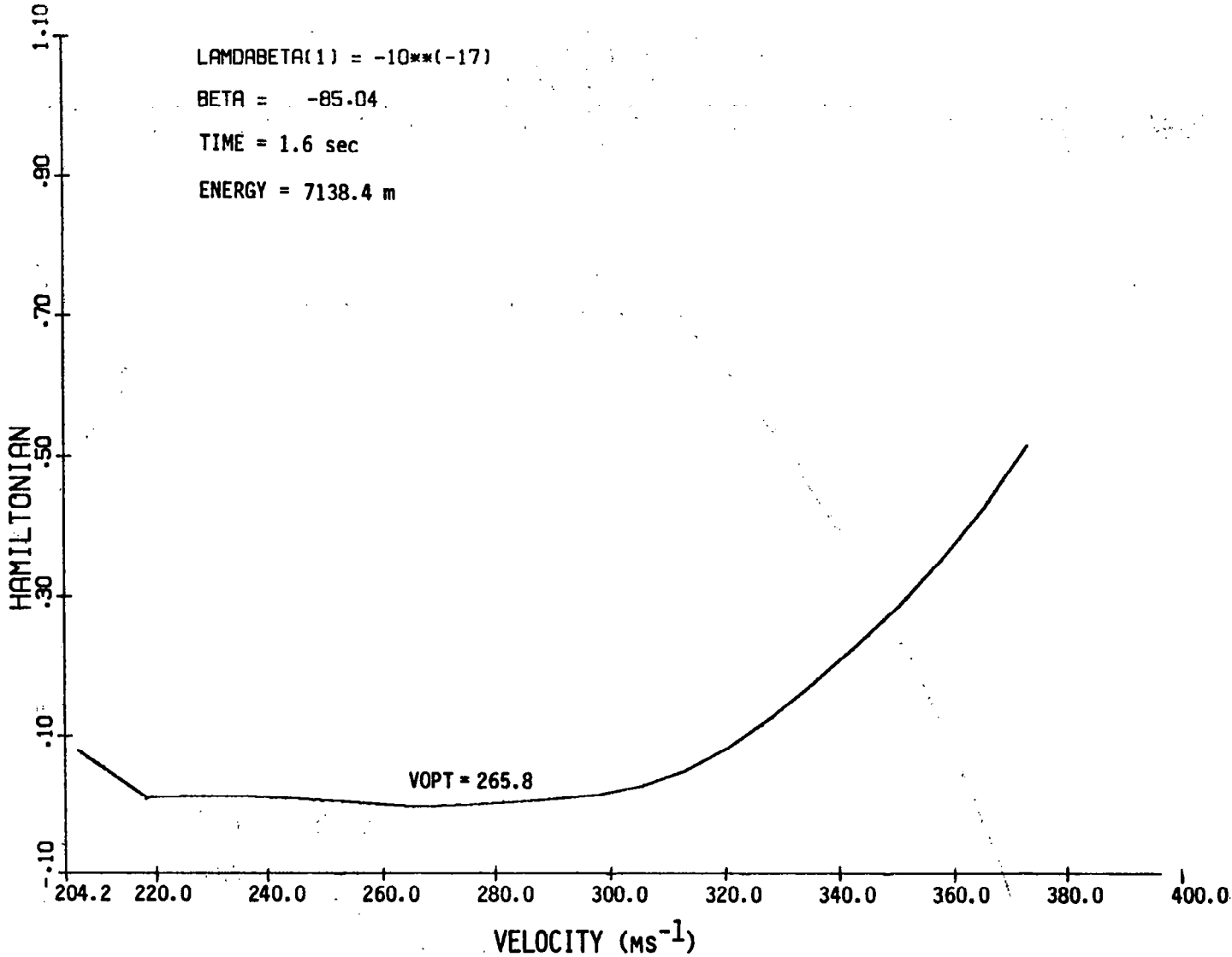


Figure A5.1.31 F(V) during first zoom dive

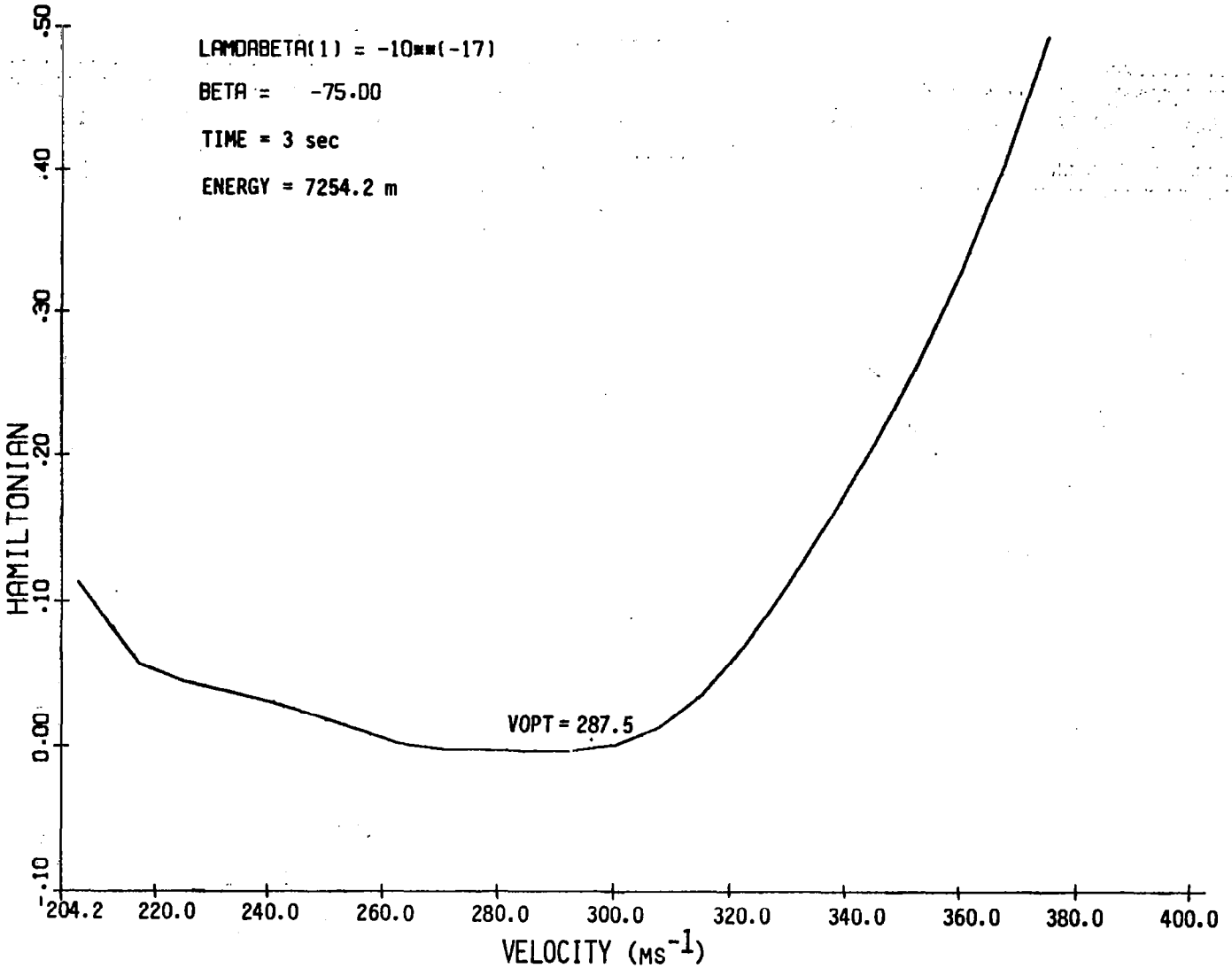


Figure A5.1.32 F(V) during first zoom dive

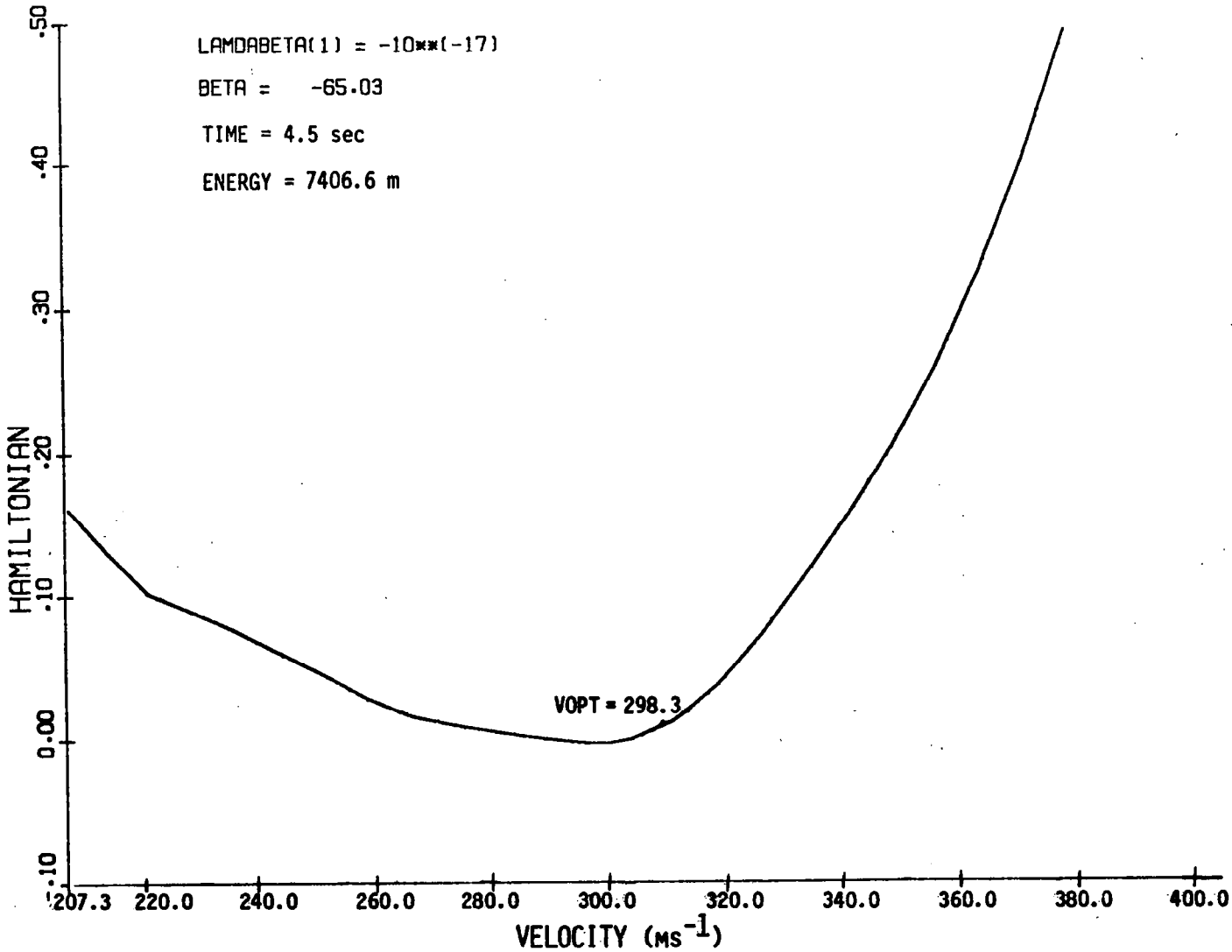


Figure A5.1.33 F(V) towards the end of the first zoom dive

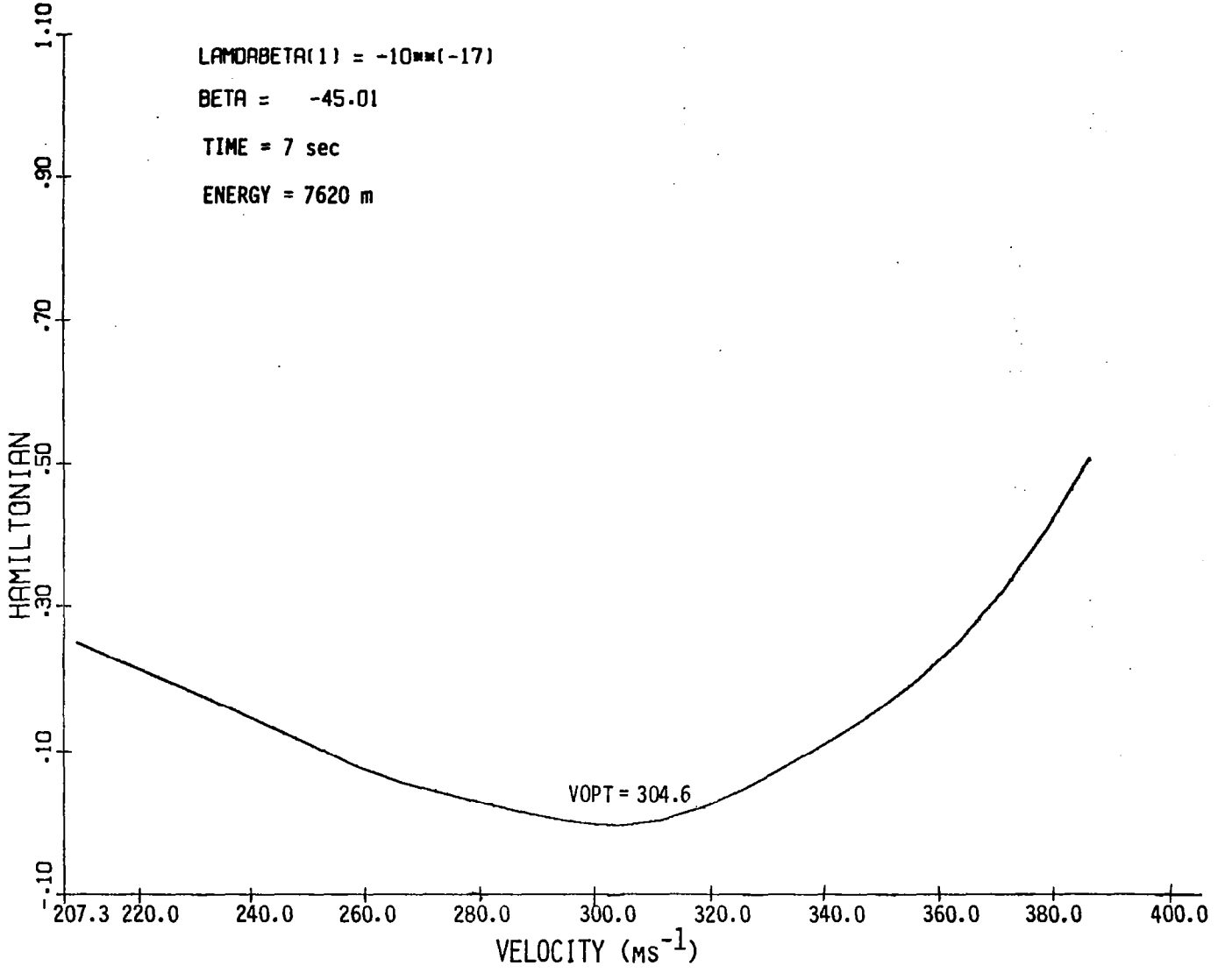


Figure A5.1.34 F(V) between the two zoom dives

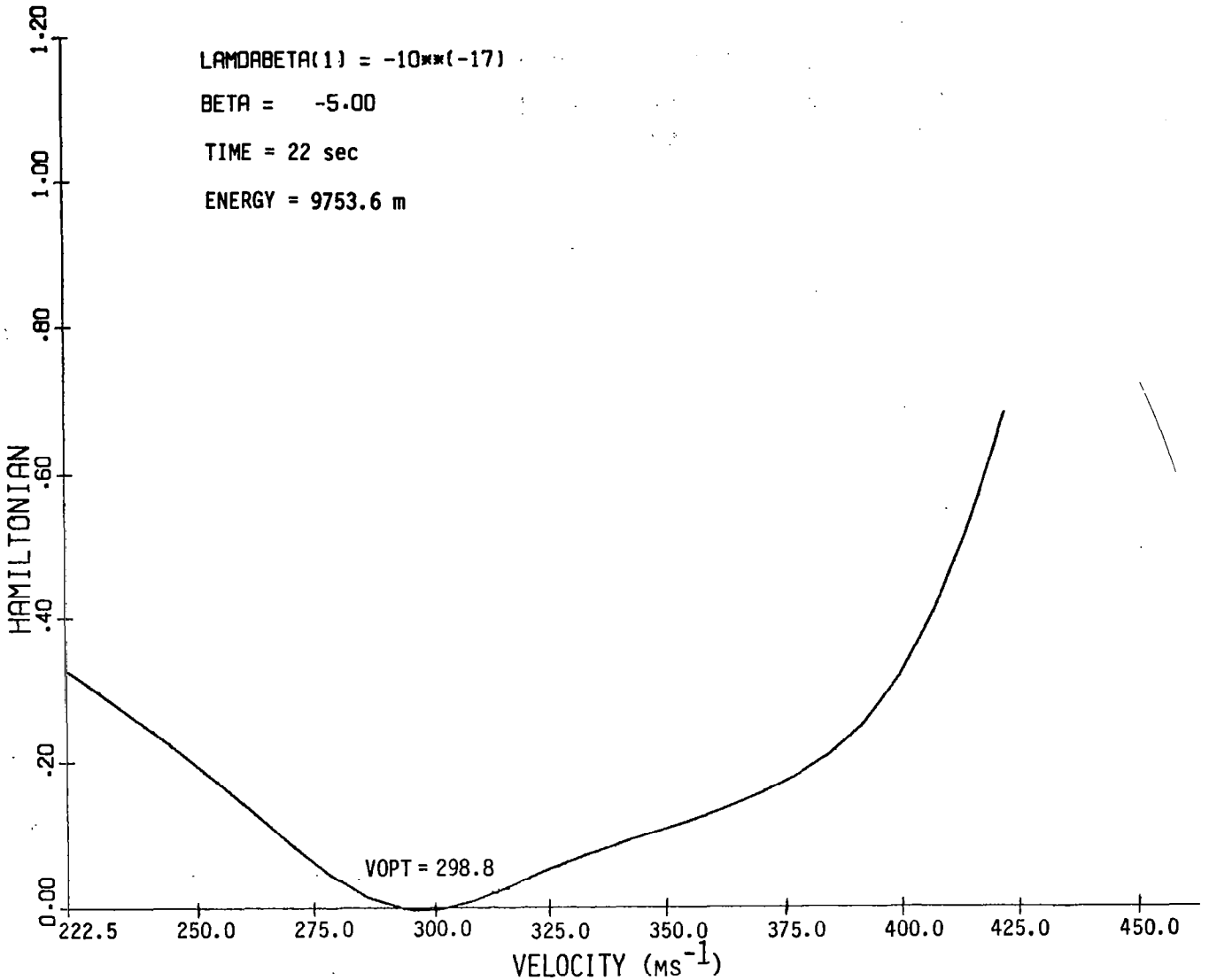


Figure A5.1.35 F(V) between the two zoom dives

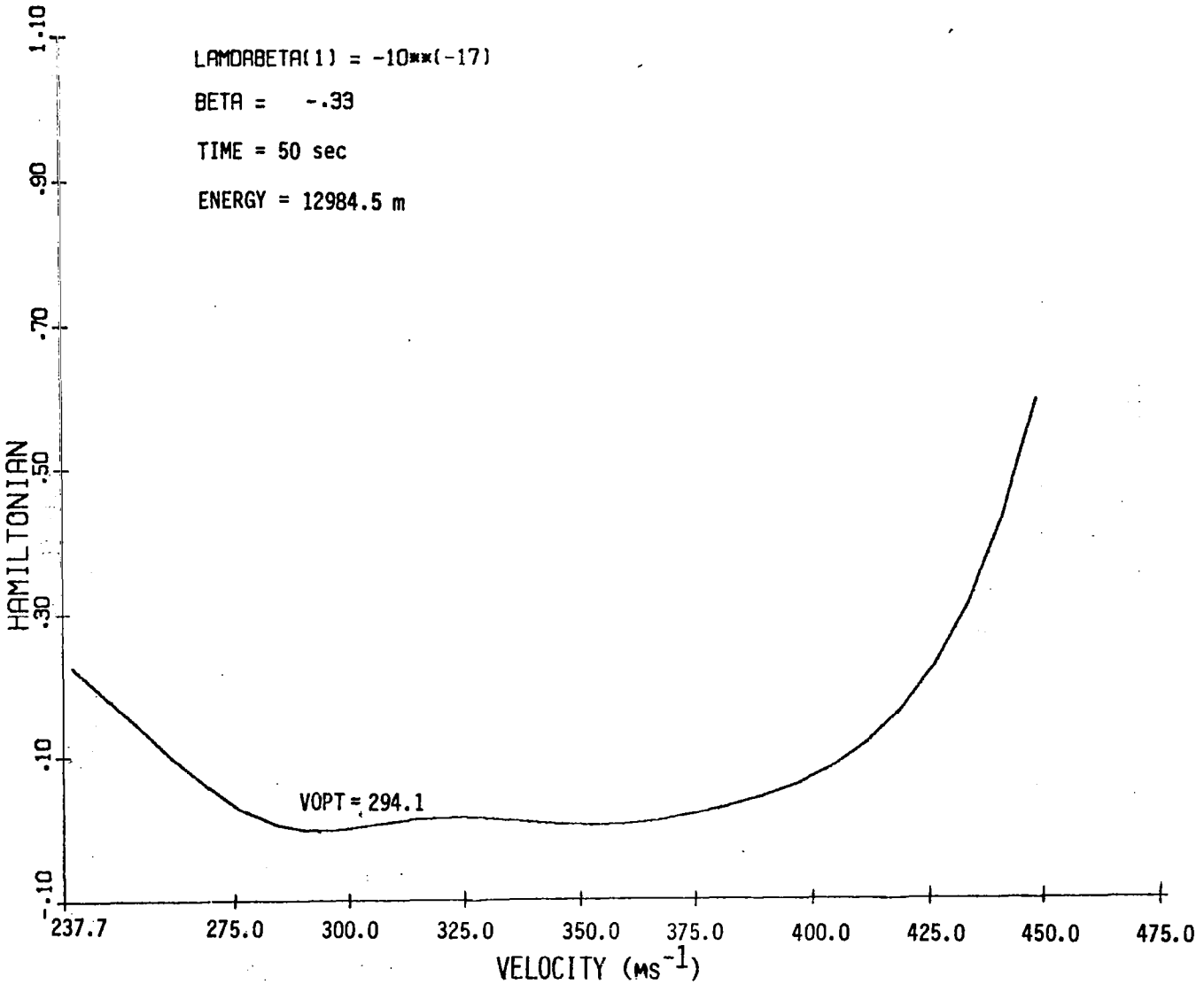


Figure A5.1.36 F(V) at the beginning of the second zoom dive

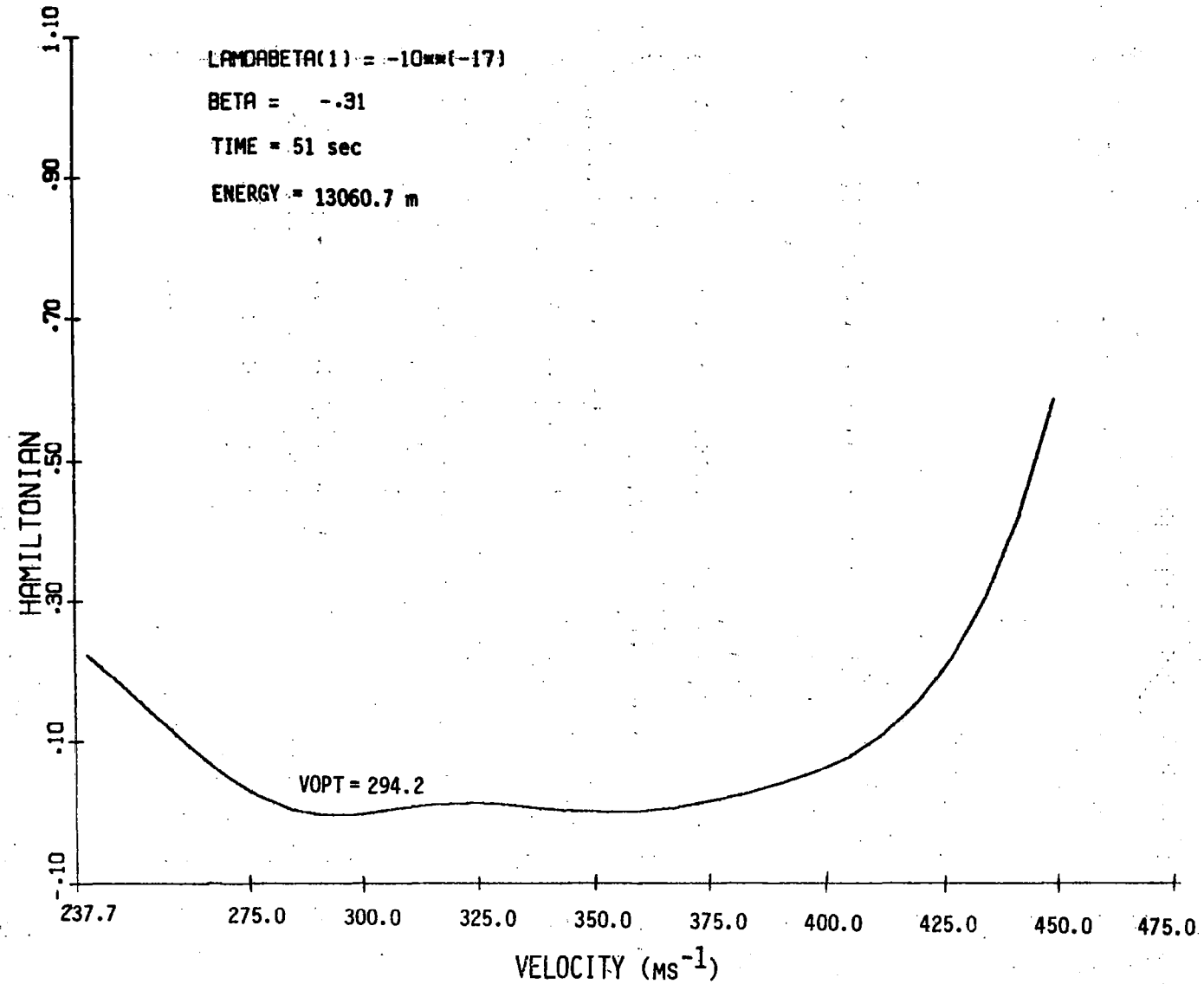


Figure A5.1.37 F(V) during the second zoom dive

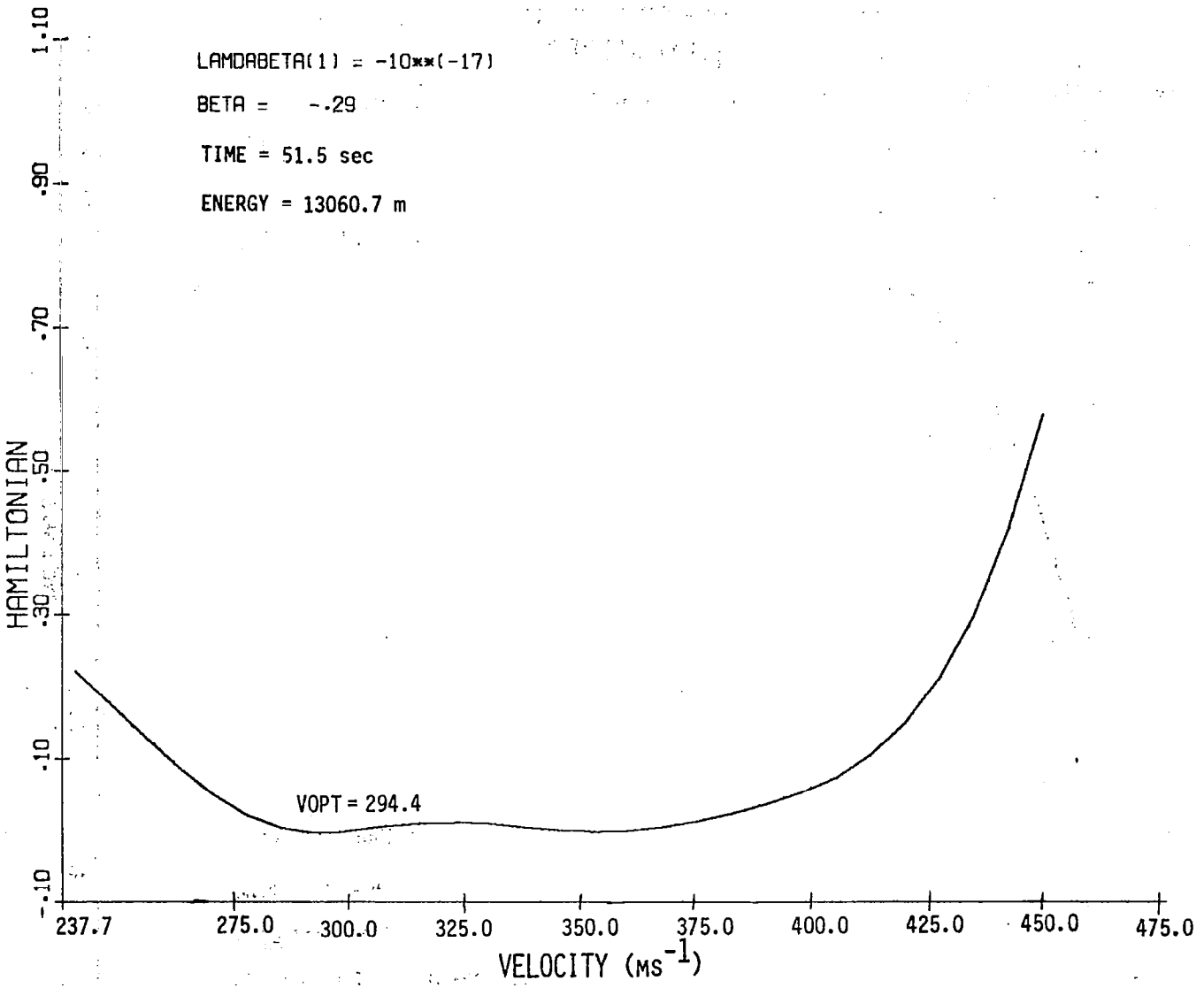


Figure A5.1.38 F(V) during the second zoom dive

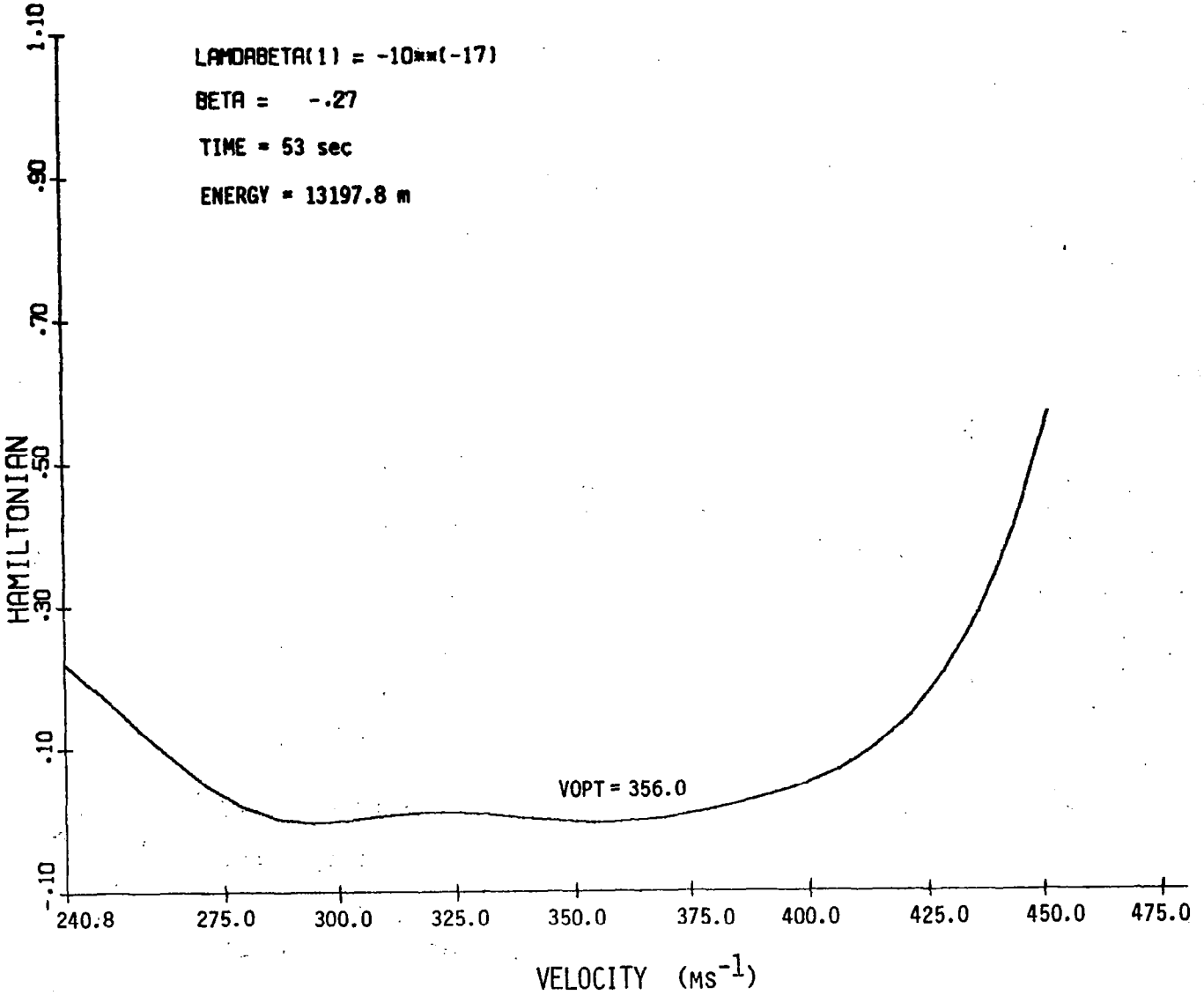


Figure A5.1.39 F(V) during the second zoom dive

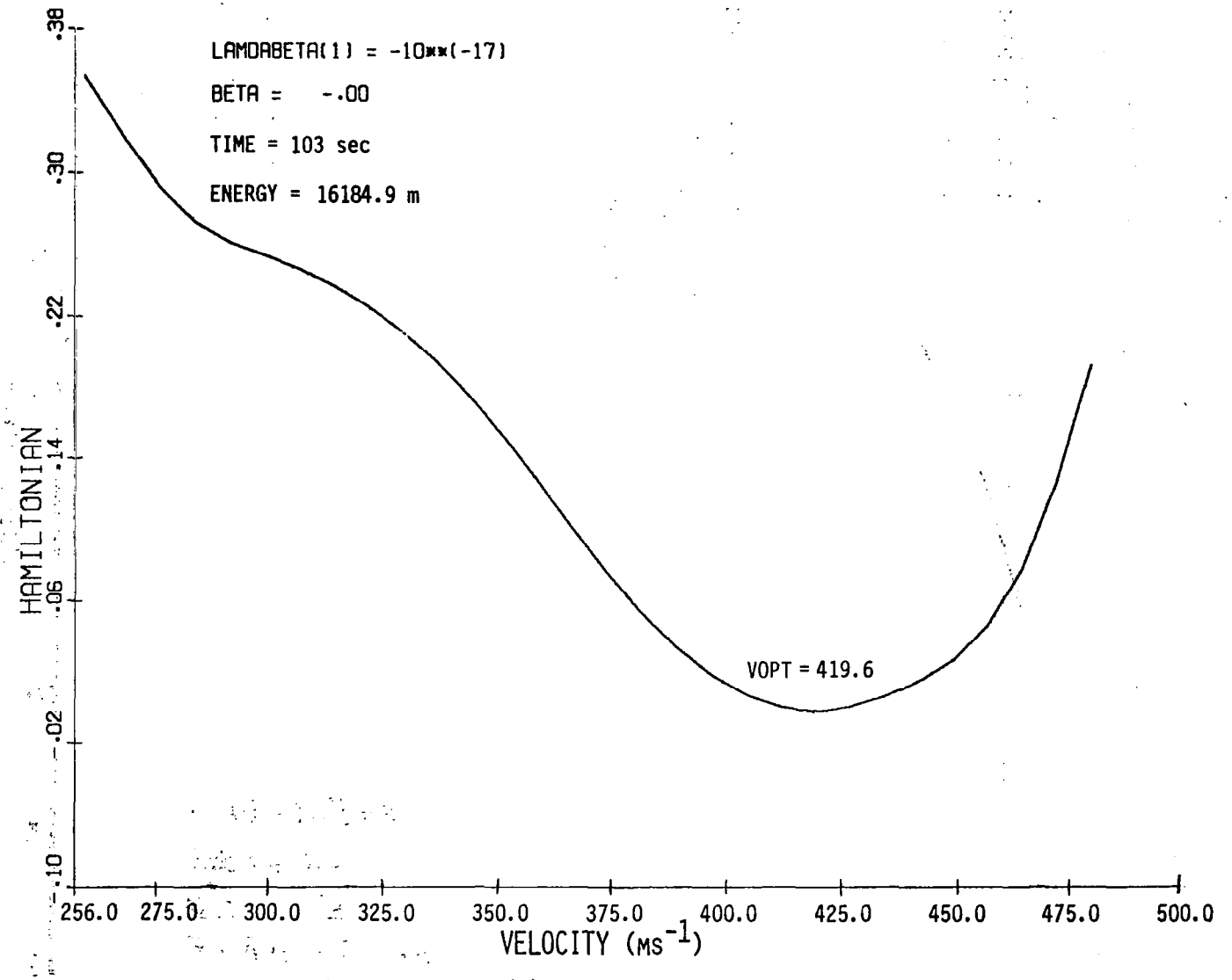


Figure A5.1.40 F(V) along $\Delta\beta = 0$ flight path

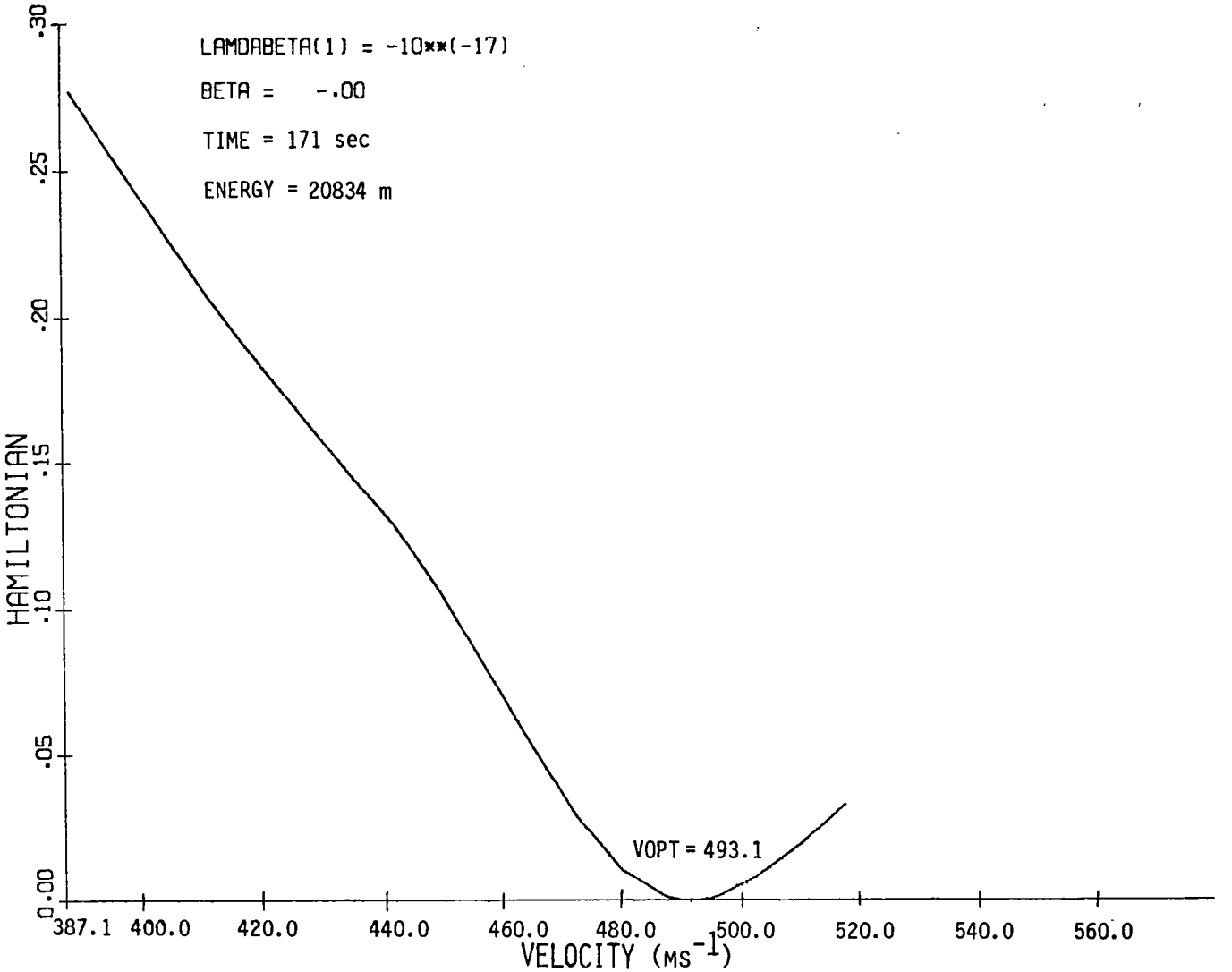


Figure A5.1.41 F(V) along $\Delta\beta = 0$ path

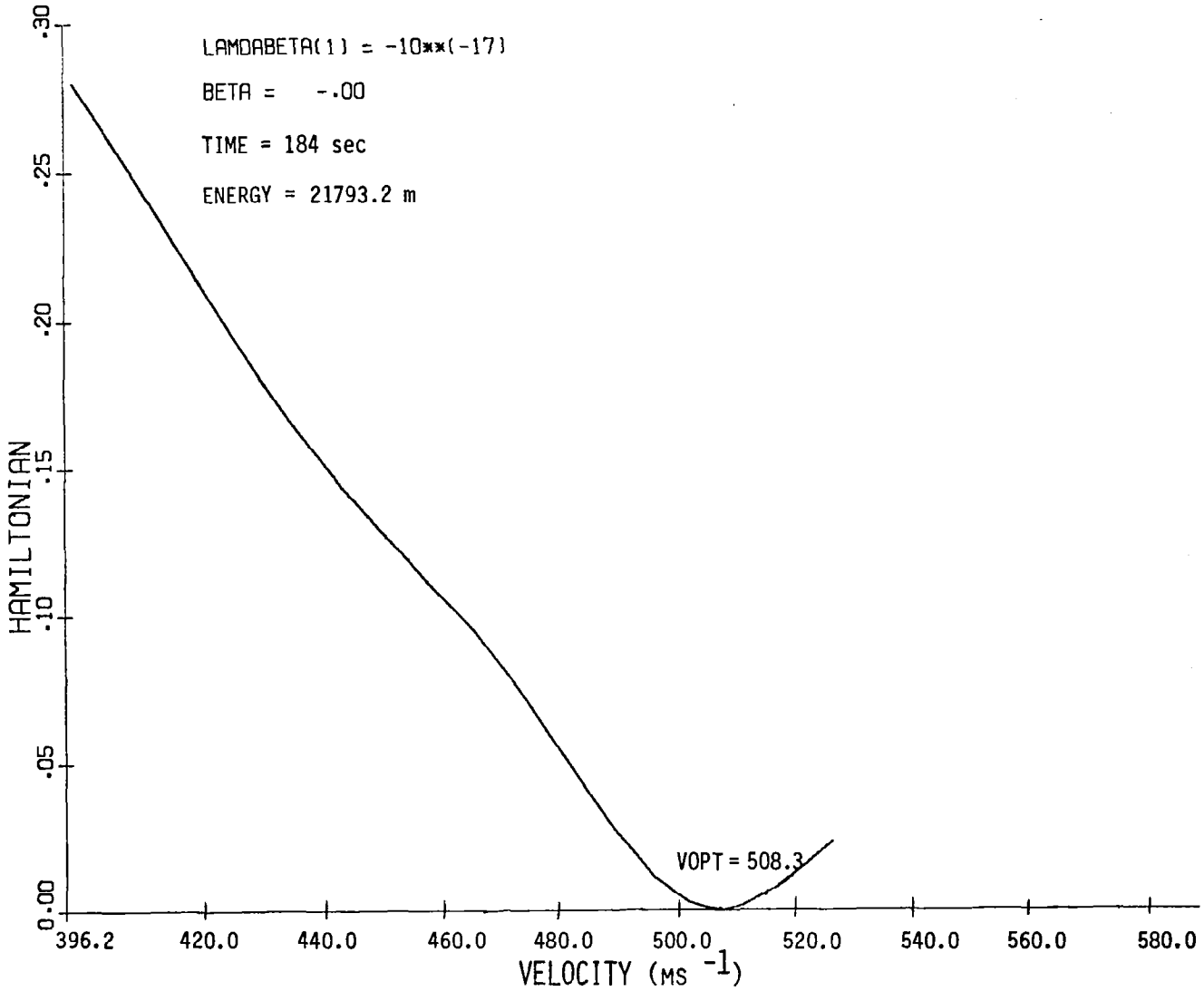


Figure A5.1.42 F(V) along $\Delta\beta = 0$ path

APPENDIX 5.2
HIGHER ORDER APPROXIMATION FOR h-LAYER SOLUTION

The exact solution of the α,γ controls for the h-boundary layer requires solving three simultaneous, nonlinear equations for α,γ and a Lagrangian multiplier λ. These equations are

(2.1) $$0 = (L_\alpha \alpha + T \sin\alpha) - mg \cos\gamma$$

(respectively equation (5.6.2) of Section 5.6)

(2.2) $$0 = \left(\frac{V}{V_c} + \lambda mg\right) - [1 + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D_0 - \alpha^2 \eta L_\alpha) + \lambda(L_\alpha \alpha + T \sin\alpha)]\cos\gamma$$

(respectively (5.6.6))

(2.3) $$0 = -\lambda_E^* \frac{V}{mg} (T \sin\alpha + 2\eta L_\alpha \alpha) + \lambda(L_\alpha + T \cos\alpha)$$

(respectively (5.6.7)).

Note that the validity of equation (2.3) depends on the optimal α not occurring on one of its constraints (i.e., $0 < \alpha < \alpha_s$ must be true).

We assume a power series expansion for $\cos\gamma$ and λ which solves (2.2), (2.3) simultaneously. That is, suppose that if $\cos\gamma = c(\alpha)$ and $\lambda = \lambda(\alpha)$ solve (2.2), (2.3), then

(2.4) $$c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + \ldots$$

(2.5) $$\lambda(\alpha) = \lambda_0 + \lambda_1\alpha + \lambda_2\alpha^2 + \ldots$$

We will solve for c_i, λ_i up to $i = 2$.

For $i = 0$, equations (2.2) and (2.3) become

(2.6) $$0 = \left(\frac{V}{V_c} + \lambda_0 mg\right) - [1 + \lambda_E^* \frac{V}{mg} (T - D_0)]c_0$$

(2.7) $$0 = \lambda_0(L_\alpha + T)$$

Hence, we have

(2.8) $$\lambda_0 = 0$$

(2.9) $$c_0 = \frac{\frac{V}{V_c}}{1 + \lambda_E^* \frac{V}{mg}(T - D_0)}$$

For i = 1, equations (2.2) and (2.3) become

(2.10) $$0 = \lambda_1 mg - [1 + \lambda_E^* \frac{V}{mg}(T - D_0)]c_0$$

(2.11) $$0 = -\lambda_E^* \frac{V}{mg}(T + 2\eta L_\alpha) + \lambda_1(L_\alpha + T)$$

Hence, we have

(2.12) $$\lambda_1 = \frac{\lambda_E^* \frac{V}{mg}(T + 2\eta L_\alpha)}{(L_\alpha + T)}$$

(2.13) $$c_1 = \frac{\lambda_E^* V(T + 2\eta L_\alpha)}{(L_\alpha + T)(1 + \lambda_E^* \frac{V}{mg}(T - D_0))}$$

For i = 2, equations (2.2) and (2.3) become

(2.14) $$0 = \lambda_2 mg - [1 + \lambda_E^* \frac{V}{mg}(T - D_0)]c_2 - [-\lambda_E^* \frac{V}{mg}(\frac{T}{2} + \eta L_\alpha) + \lambda_1(L_\alpha + T)]c_0$$

(2.15) $$0 = \lambda_2(L_\alpha + T)$$

Hence, we have

(2.16) $$\lambda_2 = 0$$

Equation (2.14) becomes after substituting λ_1 from (2.12)

$$(2.17) \qquad 0 = [1 + \lambda_E^* \frac{V}{mg} (T - D_0)]c_2 + [\lambda_E^* \frac{V}{mg} (\frac{T}{2} + \eta L_\alpha)]c_0$$

Hence, we have

$$(2.18) \qquad c_2 = \frac{-\left(\frac{V}{V_c}\right)\lambda_E^* \frac{V}{mg} (\frac{T}{2} + \eta L_\alpha)}{(1 + \lambda_E^* \frac{V}{mg} (T - D_0))^2}$$

The preceding perturbation series for $\gamma(\alpha)$ is used to solve for α from (2.1) as follows. Substitution of (2.4) in (2.1) gives

$$(2.19) \qquad 0 = L_\alpha \alpha + T \sin\alpha - mg(c_0 + c_1\alpha + c_2\alpha^2)$$

To be consistent, we should expand $\sin\alpha$ to second order in (2.19) and solve

$$(2.20) \qquad 0 = L_\alpha \alpha + T\alpha - mg(c_0 + c_1\alpha + c_2\alpha^2)$$

We could solve (2.2) directly by quadratic formula. Instead we will expand α in a perturbation series with respect to the parameter

$$(2.21) \qquad \varepsilon = \frac{mg}{L_\alpha + T}$$

Thus, (2.20) becomes

$$(2.22) \qquad 0 = \alpha - \varepsilon(c_0 + c_1\alpha + c_2\alpha^2)$$

where we assume $\alpha(\varepsilon)$ is given as

$$(2.23) \qquad \alpha = \alpha_0 + \varepsilon\alpha_1 + \varepsilon^2\alpha_2 + \ldots$$

For $i = 0$, (2.22) becomes

$$(2.24) \qquad 0 = \alpha_0$$

which gives α_0 directly.

For $i = 1$, (2.22) becomes

$$0 = \alpha_1 - c_0 \tag{2.25}$$

which gives α_1 as

$$\alpha_1 = c_0 \tag{2.26}$$

For $i = 2$, (2.22) becomes

$$0 = \alpha_2 - (c_1\alpha_1) \tag{2.27}$$

which gives

$$\alpha_2 = c_1c_0 \tag{2.28}$$

Thus, to second order in α we obtain

$$\alpha \simeq c_0\varepsilon + c_1c_0\varepsilon^2 \tag{2.29}$$

In terms of the aerodynamic coefficients (2.29) gives α as

$$\alpha \simeq \frac{\frac{V}{V_c}\,mg}{[1 + \lambda_E^* \frac{V}{mg}(T - D_0)][L_\alpha + T]} + \frac{\frac{V}{V_c}(mg)^2\lambda_E^* V(T + 2\eta L_\alpha)}{[1 + \lambda_E^* \frac{V}{mg}(T - D_0)]^2[L_\alpha + T]^3} \tag{2.30}$$

To obtain γ we may solve (2.1) directly for γ, using the α from (2.30) or we may use the series expansion for $\cos\gamma$ in (2.4) and solve for γ from that. Ignoring terms with $O(\varepsilon^3)$ dependence would give

$$\gamma \simeq \cos^{-1}\left[\left(\frac{L_\alpha + T}{mg}\right)\alpha\right] \tag{2.31}$$

where α in (2.31) is given by (2.30).

Note that at this level there is no reason to work with γ rather than $\cos\gamma$, and one may as well avoid $\cos^{-1}$.

The two things we need from the h-boundary layer are γ^* (or $\cos\gamma^*$) and λ_h^*. We have obtained the former in (2.31). We now obtain λ_h^*. The adjoint λ_h^* is such that

$$(2.32) \quad 0 = 1 - \frac{V}{mg}\cos\gamma^* + \lambda_E^* \frac{V}{mg}(T\cos\alpha^* - D_0 - \eta L_\alpha(\alpha^*)^2) + \lambda_h^* V \sin\gamma^*$$

(respectively equation (5.6.9)).

By rewriting equation (2.2) we obtain

$$(2.33) \quad \{1 - \frac{V}{V_c}\cos\gamma + \lambda_E \frac{V}{mg}(T\cos\alpha - D_0 - \alpha^2\eta L_\alpha)\}\cos\gamma$$

$$= (1 - \cos^2\gamma)(\frac{V}{V_c} + \lambda mg) + \lambda(mg\cos\gamma - L_\alpha\alpha - T\sin\alpha)\cos\gamma$$

Using equation (2.1) we can simplify (2.33) to obtain

$$(2.34) \quad \{1 - \frac{V}{V_c}\cos\gamma + \lambda_E \frac{V}{mg}(T\cos\alpha - D_0 - \alpha^2\eta L_\alpha)\}\cos\gamma = \sin^2\gamma(\frac{V}{V_c} + \lambda mg)$$

comparing (2.34) to (2.32) we see that λ and λ_h^* are related as follows:

$$(2.35) \quad \lambda_h^* = -\frac{\tan\gamma^*}{V}(\frac{V}{V_c} + \lambda^* mg)$$

where λ^* is obtained from (2.5) with α^* substituted from (2.30). Note that λ^* is given as

$$(2.36) \quad \lambda^* = \lambda_1\alpha^*$$

and from (2.12) we have

$$(2.37) \quad \lambda^* = \left\{\frac{\lambda_E^* \frac{V}{mg}(T + 2\eta L_\alpha)}{L_\alpha + T}\right\}\alpha^*$$

and hence λ_h^* is given by

$$(2.38) \qquad \lambda_h^* = -\frac{\tan\gamma^*}{V}\left(\frac{V}{V_c} + \frac{\lambda_E^* V(T + 2\eta L_\alpha)}{(L_\alpha + T)}\alpha^*\right)$$

Thus, we may use either (2.38) or (2.32) directly to obtain λ_h^* from γ^* and α^*. The direct computation of λ_h^* from (2.32) may be more accurate but (2.38) seems to give a better idea of how λ_h^* behaves in terms of V, γ^*, α^*. Note that the only error in (2.38) comes from the approximation of λ^*, and this approximation has an error of $O([\alpha^*]^3)$.

Constraints. The next possibility to check is whether or not α or γ takes a value on a constraint. To denote possible constraints, define γ_s as the positive solution γ of

$$(2.39) \qquad 0 = L_\alpha\alpha_s + T\sin\alpha_s - mg\cos\gamma$$

if such a solution exists, i.e., if we have

$$(2.40) \qquad \frac{L_\alpha\alpha_s + T\sin\alpha_s}{mg} \leq 1$$

Likewise, define α_m as the solution α of

$$(2.41) \qquad 0 = L_\alpha\alpha + T\sin\alpha - mg$$

Analysis of the original optimization problem, to minimize (for $\gamma > 0$) the function

$$(2.42) \qquad \frac{1 - \frac{V}{V_c}\cos\gamma + \lambda_E^*\frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha\alpha^2)}{V\sin\gamma}$$

(respectively, equation (5.6.4)), subject to the constraints

$$(2.43) \qquad 0 < \gamma < \pi$$

$$(2.44) \qquad 0 \leq \alpha \leq \alpha_s$$

and (2.1), shows that as $\gamma \to 0$, the function in (2.42) tends to $+\infty$ as $\frac{1}{\gamma}$. Hence,

we must have

$$\alpha < \alpha_m \tag{2.45}$$

where α_m is defined by (2.40) above. Note that $\alpha_m \leq \alpha_s$ or $\alpha_m \geq \alpha_s$ may occur.

Thus, the only constrained values of (α,γ) are due to the constraints on α in (2.44).

If $\alpha_s < \alpha_m$, then we check the possibility of

$$-\lambda_h^s = \frac{1 - \frac{V}{V_c}\cos\gamma_s + \lambda_E^* \frac{V}{mg}(T\cos\alpha_s - D_0 - \eta L_\alpha \alpha_s^2)}{V \sin\gamma_s} \tag{2.46}$$

being optimal. Likewise, we must check the possibility of

$$-\lambda_h^0 = \frac{1 + \lambda_E^* \frac{V}{mg}(T - D_0)}{V} \tag{2.47}$$

being optimal (the $\alpha = 0$, $\gamma = \pi/2$ solution).

The optimality test for these constraints simply involves comparing λ_h^s and λ_h^0 to λ_h^* for the unconstrained problem. The comparison goes as follows:

Case 1. $\alpha^* \leq 0$ or $\alpha_s \leq \alpha^*$, either λ_h^0 or λ_h^s is optimal. The optimal case corresponds to the larger or the two (remember $-\lambda_h^0$ or $-\lambda_h^s$ is minimum).

Case 2. $0 < \alpha^* < \alpha_s$, either λ_h^0, λ_h^s or λ_h^* is optimal. The optimal case corresponds to the largest of the three.

h-Boundary Layer Algorithm

1. Calculate α_m from (2.41).
2. Calculate α^* from (2.30) (second order approximation) or from the first order approximation

$$\alpha^* \simeq \frac{\frac{V}{V_c} mg}{[1 + \lambda_E^* \frac{V}{mg}(T - D_0)][L_\alpha + T]} \tag{2.48}$$

or, if you prefer, by solving the nonlinear equation

$$(2.49) \qquad 0 = L_\alpha \alpha + T \sin\alpha - \frac{\frac{V}{V_c} mg}{1 + \lambda_E^* \frac{V}{mg} (T - D_0)}$$

by a Newton-Raphson method.

3. If $\alpha_m \leq \alpha_s$, then go to 6.
4. If $\alpha^* < 0$ or $\alpha_s < \alpha^*$ go to 7.
5. Calculate λ_h^0, λ_h^*, λ_h^s: the largest corresponds to the optimal solution,

   $\lambda_h^0 \rightarrow \alpha = 0, \gamma = \pi/2$

   $\lambda_h^* \rightarrow \alpha^*, \gamma^*$

   $\lambda_h^s \rightarrow \alpha_s, \gamma_s$

   Go to 8.
6. Calculate α^*, if $\alpha^* < 0$ then λ_h^0 is optimal and $\alpha = 0$, $\gamma = \pi/2$. If $\alpha^* > 0$ then calculate λ_h^0, λ_h^* and choose the larger--this gives the optimal.
   Go to 8.
7. Calculate λ_h^0, λ_h^s. The larger corresponds to the optimal solution. Go to 8.
8. If $h < h(E)$, keep $\gamma > 0$ as it is ($\gamma = \gamma_s$, $\pi/2$ or γ^*). If $h > h(E)$, take $-\gamma$ instead.

STOP

APPENDIX 5.3
HIGHER ORDER APPROXIMATION OF γ-LAYER SOLUTION

This appendix presents a higher order approximation of the γ-boundary layer calculation given in Section 5.7. The principal improvement is to use a Taylor series expansion to determine the perturbation of the angle-of-attack from the value which maintains a steady flight path angle. By calculating $\delta\alpha$, the perturbation of the angle-of-attack α from its steady state value α_0 instead of calculating the angle-of-attack α directly, we will obtain better numerical behavior of the solution.

To define the problem more precisely, we must make some preliminary definitions. Let $L(\alpha)$ be the function of the angle-of-attack α defined by

$$(3.1) \qquad L(\alpha) = 1 - \frac{V}{V_c}\cos\gamma + \gamma^*_E \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2) + \lambda^*_h V \sin\gamma$$

and let $f(\alpha)$ be the function of α defined by

$$(3.2) \qquad f(\alpha) = L_\alpha \alpha + T\sin\alpha - mg\cos\gamma$$

Note that the functions L and f also depend on the flight path angle γ. The optimization problem is to minimize the ratio L/f if $\gamma < \gamma^*(E,h)$ and to maximize this ratio if $\gamma > \gamma^*(E,h)$, where $\gamma^*(E,h)$ denotes the pseudocontrol value of γ calculated in the h-boundary layer. In the minimization we are to restrict values of α such that $f(\alpha) > 0$; likewise, in the maximization we are to restrict α so that $f(\alpha) < 0$. In addition there are inequality constraints on α, namely

$$(3.3) \qquad 0 \le \alpha \le \alpha_s$$

Let us define the steady state value of α as the value of α, denoted α_0, which solves the equation

$$(3.4) \qquad 0 = f(\alpha_0)$$

That is, α_0 is the angle-of-attack which maintains γ in steady state. Note that

α_0 depends on γ (as well as E and h through the aerodynamic coefficients L_α and T). As pointed out in Appendix 4.2 on the existence of solutions to boundary layer calculations, a solution to the optimization problem will exist in this case with $\alpha > \alpha_0$ for $\gamma < \gamma^*$ and $\alpha < \alpha_0$ for $\gamma > \gamma^*$. The former case corresponds to $f(\alpha) > 0$ and the latter case corresponds to $f(\alpha) < 0$. Thus, in either the minimization or the maximization it will be necessary for α to be found in the interval $\alpha_0 < \alpha \leq \alpha_s$ (for the minimization) or in the interval $0 \leq \alpha < \alpha_0$ (for the maximization). In each case, the optimal α is either an unconstrained value or it is α_s (in the case of the minimization) or 0 (in the case of the maximization).

As we pointed out in Section 5.7, when $E < E_c$ and $\gamma > \gamma^*$ the optimal α is 0; similarly, when $E > E_c$ and $\gamma < \gamma^*$ the optimal α is α_s. Thus, we will consider only the other two cases when $E > E_c$ and $\gamma > \gamma^*$ or when $E < E_c$ and $\gamma < \gamma^*$. To determine the optimal α in these cases we set the derivative of $L(\alpha)/f(\alpha)$ equal to 0 in order to determine the unconstrained values of α and then compare these values to the corresponding constrained values of α. Setting the derivative of $L(\alpha)/f(\alpha)$ equal to 0 gives the equation

$$f(\alpha)L'(\alpha) - f'(\alpha)L(\alpha) = 0 \tag{3.5}$$

To obtain an approximate expression for the solution α_u (unconstrained α) of (3.5) we expand $L(\alpha)$ and $f(\alpha)$ in a Taylor series around α_0. Thus, we obtain

$$L(\alpha) = L(\alpha_0) + L'(\alpha_0)\delta\alpha + \tfrac{1}{2}L''(\alpha_0)(\delta\alpha^2) \tag{3.6}$$

and

$$f(\alpha) = f'(\alpha_0)\delta\alpha + \tfrac{1}{2}f''(\alpha_0)(\delta\alpha^2) \tag{3.7}$$

Note that we have neglected terms higher than second order. Substituting (3.6) and (3.7) into (3.5) and collecting terms gives the following quadratic equation for α.

$$f'(\alpha_0)L(\alpha_0) + f''(\alpha_0)L(\alpha_0)(\delta\alpha) + \tfrac{1}{2}(f''(\alpha_0)L'(\alpha_0) - f'(\alpha_0)L''(\alpha_0))(\delta\alpha^2) = 0 \tag{3.8}$$

The solution of this equation is given by the quadratic formula

$$(3.9)\qquad \delta\alpha = \frac{-f''(\alpha_0)L(\alpha_0) \pm \sqrt{\{(f''(\alpha_0)L(\alpha_0))^2 - 2f'(\alpha_0)L(\alpha_0)(f''(\alpha_0)L'(\alpha_0) - f'(\alpha_0)L''(\alpha_0))\}}}{\{f''(\alpha_0)L'(\alpha_0) - f'(\alpha_0)L''(\alpha_0)\}}$$

To determine which root to choose we must calculate the derivatives of f and L from the original expressions (3.1) and (3.2) for f and L. Thus, we obtain

$$(3.10)\qquad L(\alpha_0) = 1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha_0 - D_0 - \eta L_\alpha \alpha_0^2) + \lambda_h^* V \sin\gamma$$

$$(3.11)\qquad L'(\alpha_0) = -\lambda_E^* \frac{V}{mg}(T\sin\alpha_0 + 2\eta L_\alpha \alpha_0)$$

$$(3.12)\qquad L''(\alpha_0) = -\lambda_E^* \frac{V}{mg}(T\cos\alpha_0 + 2\eta L_\alpha)$$

$$(3.13)\qquad f'(\alpha_0) = L_\alpha + T\cos\alpha_0$$

$$(3.14)\qquad f''(\alpha_0) = -T\sin\alpha_0$$

Substitution of (3.10) - (3.14) into (3.9) gives us

$$(3.15)\qquad \delta\alpha = -\frac{TL}{K}\left\{\sin\alpha_0 \pm \sqrt{\sin^2\alpha_0 + \frac{2(L_\alpha + T\cos\alpha_0)}{TL}K}\right\}$$

where K is given by

$$(3.16)\qquad K = -\frac{\lambda_E^* V}{mg}(T^2 + 2\eta L_\alpha T(\alpha_0\sin\alpha_0 + \cos\alpha_0) = L_\alpha T\cos\alpha_0 + L_\alpha^2 2\eta)$$

and L is $L(\alpha_0)$. Note that K has the opposite sign of λ_E^*. Also note that L, T, L_α, η, $\cos\alpha_0$, $\sin\alpha_0$ are all positive. Knowledge of the signs of these quantities will enable us to determine which root to take in (3.15).

Suppose that $E < E_c$ and $\gamma < \gamma^*$. In this case we must have $\delta\alpha > 0$, Since $E < E_c$, the adjoint λ_E^* is negative and thus the coefficient K is positive. Thus, in this case the unconstrained value for $\delta\alpha$ is given by

$$(3.17)\qquad \delta\alpha = -\frac{TL}{K}\left\{\sin\alpha_0 + \sqrt{\sin^2\alpha_0 - 2(L_\alpha + T\cos\alpha_0)\frac{K}{TL}}\right\}$$

On the other hand, if $E > E_c$ and $\gamma > \gamma^*$, then we must have $\delta\alpha < 0$. Moreover, the coefficient K is negative in this case. Unfortunately, this means that both roots of (3.15) are negative and we cannot immediately distinguish which is the correct root. Moreover, for K negative it is possible to obtain complex roots in (3.15) and we must be able to handle this case adequately to obtain a real value for $\delta\alpha$. What follows is an algorithm for calculating $\delta\alpha$ in the various cases of E and γ.

γ-Boundary Layer Calculation: Algorithm

1. If $E < E_c$ and $\gamma > \gamma^*$, let $\alpha = 0$ and stop.
2. If $E > E_c$ and $\gamma < \gamma^*$, let $\alpha = \alpha_s$ and stop.
3. If $E < E_c$ and $\gamma < \gamma^*$, calculate $\delta\alpha$ from (3.17) and go to step 5.
4. If $E > E_c$ and $\gamma > \gamma^*$, calculate both roots from (3.15) and go to step 6.
5. If $\alpha_0 + \delta\alpha > \alpha_s$, let $\alpha = \alpha_s$ and stop; otherwise, go to step 7.
6. Replace complex roots by real part. If $\alpha_0 + \delta\alpha < 0$ (for <u>both</u> roots), let $\alpha = 0$ and stop. Otherwise discard any root such that $\alpha_0 + \delta\alpha < 0$ and go to step 8.
7. Let $\alpha_u = \alpha_0 + \delta\alpha$. Choose $\alpha = \alpha_u, \alpha_s$ whichever minimizes $L(\alpha)/f(\alpha)$. Stop.
8. Let $\alpha_u = \alpha_0 + \delta\alpha$ (both roots for $\delta\alpha$). Choose $\alpha = \alpha_u$ or $\alpha = 0$ whichever maximizes $L(\alpha)/f(\alpha)$. Stop.

CHAPTER 6
REAL-TIME SPT CONTROL LAW

This chapter describes the real-time SPT control law as it has been implemented and tested on the CDC 6400 computer. Section 6.1 contains a description of the control logic for different parts of the flight, namely before the cruise arc (Case I), on the cruise arc (Case II), and after the cruise arc (Case III). An estimate of the total computations required is also given. The exact computer time for each cycle of the control computation will depend on the characteristics of the flight computer. This in turn will determine the control update rate, which can vary for different control loops. In general, the computation rate for the faster layers must be kept higher than that for the slower layers in accordance with the SPT approximation. Fortunately, the computations involved at faster layers are less than those at slower layers so that the SPT algorithms of Chapter 5 are easily implemented to satisfy this requirement.

Other sections of Chapter 6 contain results of using the real-time control algorithm for an F-4 aircraft. The results are very encouraging both in terms of accuracy and computation time.

6.1 Description of Real-Time Control Logic

The purpose of real-time control logic is to compensate for changes in target relative position, velocity and heading. A change in target velocity produces a change in the intercept point, namely, t_f, $x(t_f)$ and $y(t_f)$. If the pursuer's position is on or before the cruise arc, this requires a change in cruise heading β_c and time t_c for coming off the cruise arc. It is convenient to separate the control logic into three parts depending on the position of the pursuing aircraft relative to the target.

6.1.1 Case I: Before the Cruise Arc

If the pursuer is on the energy climb path to the cruise arc and β_c is changed, a change is required in β before the cruise arc. For small β-changes, a linear control law relating σ to (E,β) can be used. This may produce small

changes in (E,h,γ) which again can be compensated by linear feedback law. The aircraft state would return to the energy climb path after the required correction in β has been made.

If the change in β is large, Parsons' (1972) energy state solution suggests the use of nonlinear control law. This requires zoom climb or dive to the max-turn-rate (MTR) locus, heading change through the appropriate amount and zoom dive or climb to the min-time energy (MTE) climb path. Boundary layer control calculations for h and γ are required to complete the zoom maneuvers. The latter are basically vertical plane maneuvers at constant thrust, requiring calculation of α to complete the zooms. However, simulation results for the F-4 have shown that it is better to turn at current energy without zooms since the zoom dynamics is quite slow (see Tables 6.2.1, 6.2.2 and 6.2.3).

6.1.2 Case II: On the Cruise Arc

Control law similar to Case I is used here except that it is much simpler in form since the linearized system is time invariant and the optimal cruise energy is constant. Thus, constant feedback law is appropriate for small changes in β_c, h and γ.

6.1.3 Case III: After the Cruise Arc

The terminal part of this case involves short range optimization problem which is beyond the scope of the present study since SPT is not valid for this case. However, for small changes in terminal conditions, a linearized solution is appropriate and this would take the same form as the linear solution in Case I.

6.1.4 Feedback Structure of the On-Line SPT Algorithm

Figure 6.1.1 shows a schematic of the on-line SPT algorithm. From a feedback control viewpoint, it is a hierarchical structure with six loops. The fastest loop, i.e., γ-loop, is at the bottom of the hierarchy and it requires inputs from all the higher loops. Since the higher loops operate at slower speeds, these inputs to the γ-loop are updated at different rates which are much less than the speed of the γ-loop. The command γ^* is provided by the h-loop and the nonlinear SPT control law computes α to follow γ^*. If the error $|\gamma - \gamma^*|$ is small, a linear feedback control law involving both h and γ is used to take into account the interaction between the two loops.

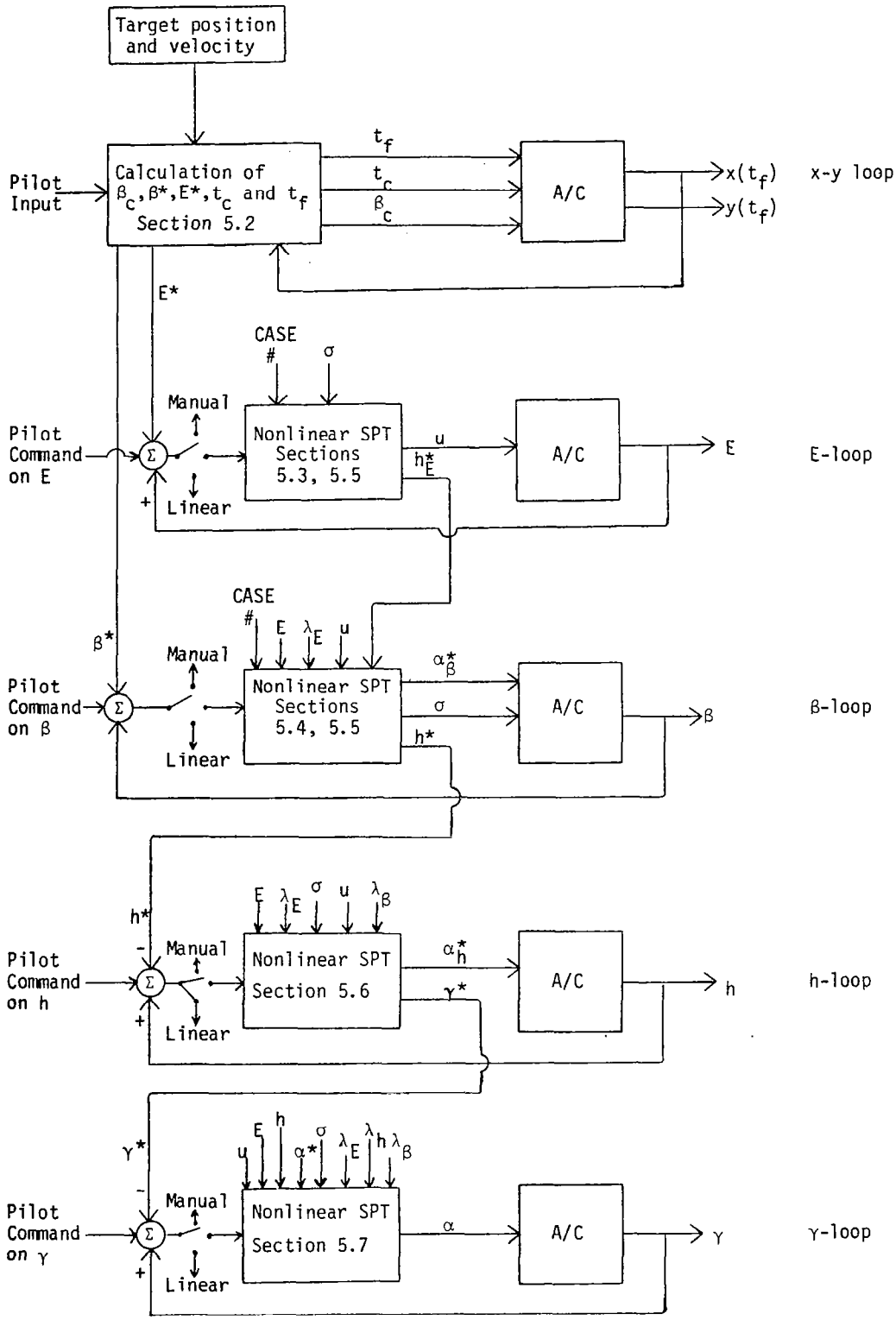


Figure 6.1.1
Hierarchical Feedback Structure of the On-line SPT Algorithm

The sampling times and the control update rates of the different loops increase as one goes down the hierarchy. This causes no problem for the SPT algorithm since the optimization problem at lower levels is simpler than the optimization problem at higher levels. In Figure 6.1.1, Case # corresponds to Cases I, II and III. The notation α_h^* denotes the optimal value of α in the h-boundary layer, which is the asymptotic value for α in the γ-boundary layer. Thus the calculations of α at higher levels serve as checks on the calculation of α at lower levels and may be used to shorten the duration of the boundary layers.

6.1.5 Real-Time Capability Assessment

In this section, we provide execution time and storage estimates for the different control loops of Figure 6.1.1 on a Texas Instruments 9900 microcomputer. TI9900 has a memory of 32K, 16 bit words and basic instruction execution times in microseconds are given in Table 6.1.1. The CPU times per iteration, nominal number iterations and storage requirements for control loops of Figure 6.1.1 are given in Table 6.1.2. The γ-loop, for example, takes 1.5 msec of CPU time per iteration for a control α update and, therefore, can be iterated up to six times for 9 msec if the measurement sampling interval for γ is, say 10 msec. Slower sampling rates for γ state would allow time for more iterations of the control loop. The control update rate for the complete on-line calculation would be determined by the radar measurement rate. Assuming 10 samples per second, a total of 100 msec are available for control computation. If the CPU time is allocated equally between all the five loops, 20 msec of computation will be available for each control loop. This implies that two iterations of the γ-loop calculations can be performed 6 times during 20 msec so that a sampling rate as high as 250/sec can be used for γ. It is clear from these figures and the numbers in Table 6.1.2 that the proposed algorithm can be easily implemented on-line using a TI9900 microcomputer.

The estimate of 27 msec for a complete control calculation of all the loops including linear feedback control implies that radar measurement rates of as high as 30 samples/second can be used. This would allow interception of highly maneuverable targets. On the other hand, if lower radar measurement rates are used, CPU time is available for further refinement of the SPT controls. In particular, the implementation of one or two iterations of the "continuation

Table 6.1.1
Execution Times for TI9900

Instruction	Execution time (microseconds) (Clock rate is 3 MHz)
Branch	
Register to register	2.67
Add (words/bytes)	
Register to register	4.67
Indirect to indexed	8.67
Multiply	
Register to register	17.33
Divide	
Register to register	41.33
Shift (left/right)	
1 bit	4.67
8 bits	9.33
Move data (words/bytes)	
Register to register	4.67
Register to directory/index	7.33
Load communications register unit (register to CRU)	
8 bits	12
16 bits	17.33
Store CRU (CRU to register)	
8 bits	14.67
16 bits	20

Table 6.1.2

TI9900 CPU and Storage Requirements for the Real-Time SPT Algorithm

	Number of average iterations required	CPU/ iteration (msec)	Average cycle time (msec)	Storage (words)
t_f - calculation	4	2	8	400
β_c - calculation	4	1	4	400
E - BL	4	1	4	210
Linear feedback around x-y solution (u,σ,α controls)	1	0.5	0.5	20
Linear feedback around E-BL (σ,α controls)	1	0.5	0.5	100
β - BL	2	2	4	500
Linear feedback around β-BL (σ,α controls)	1	0.5	0.5	100
h - BL	4	0.5	2	10
γ - BL	2	1.5	3	10
Linear feedback for h and γ (α controls)	1	0.5	0.5	400
				2150
Estimated storage for software				4000
TOTAL			27	6150

BL = boundary layer

Typical sampling interval for radar returns = 100 msec

algorithms" discussed in Chapter 3 seems feasible within a total CPU time of 100 msec. Table 6.1.2 also shows that the storage requirements are well within the 32K limit of the TI9900.

6.2 Simulation Results

In this section and the next, we describe simulation results obtained using the real-time SPT control algorithm. First a more detailed description of the actual algorithm is given since certain minor but important modifications to the SPT control law were required.

6.2.1 Climb to Cruise Arc

The optimal flight path to the cruise arc is obtained by retrieving from storage $V^*(E)$ which provides the necessary information to construct the E boundary layer. h and γ boundary layers are then added onto the E-layer solution.

6.2.1.1 E-Boundary Layer Calculations

As mentioned in Subsection 5.3.2, the necessary quantities for the E boundary layer control problem are $V^*(E)$, $h^*(E)$, $\alpha^*(E)$, $u^*(E)$ and $\lambda^*(E)$. Having determined $V^*(E)$, however, the remaining quantities are directly determined from E and $V^*(E)$.

$V^*(E)$ was determined off-line for a fine grid (1 point/61 m) of values of E, and a piecewise linear approximation was then applied. For $E < E_c$, $V^*(E)$ was determined by maximizing (5.3.10) subject to (5.3.11). The maximization was performed by exhaustive search on a grid (1 pt/(12.2 m/sec)). For $E > E_c$, $V^*(E)$ was determined by minimizing (5.3.10) subject to (5.3.11), again by exhaustive search. For $E > E_c$, it was found that $V = V_c = 590.2$ ms^{-1} was optimal.

Figure 6.2.1 shows the piecewise linear approximation for $V^*(E)$. The approximation was chosen such that its divergence from $V^*(E)$ was always less than 6.1 m/sec. We note that there is a jump from 585.2 to 590.2 ms^{-1} at $E = E_c$.

Figure 6.2.2 contains a flowchart which represents the E boundary layer control program. This flowchart may be regarded as a detailed version of Figure 5.3.1.

We note that in the last block of the flowchart the quantity $\frac{dE}{dt}$ is computed. This is the implementation of (5.3.1) with $\alpha = \alpha^*$, $V = V^*$ and $uT_{max} = T$. This calculation will not be required on board the aircraft since E would be measured

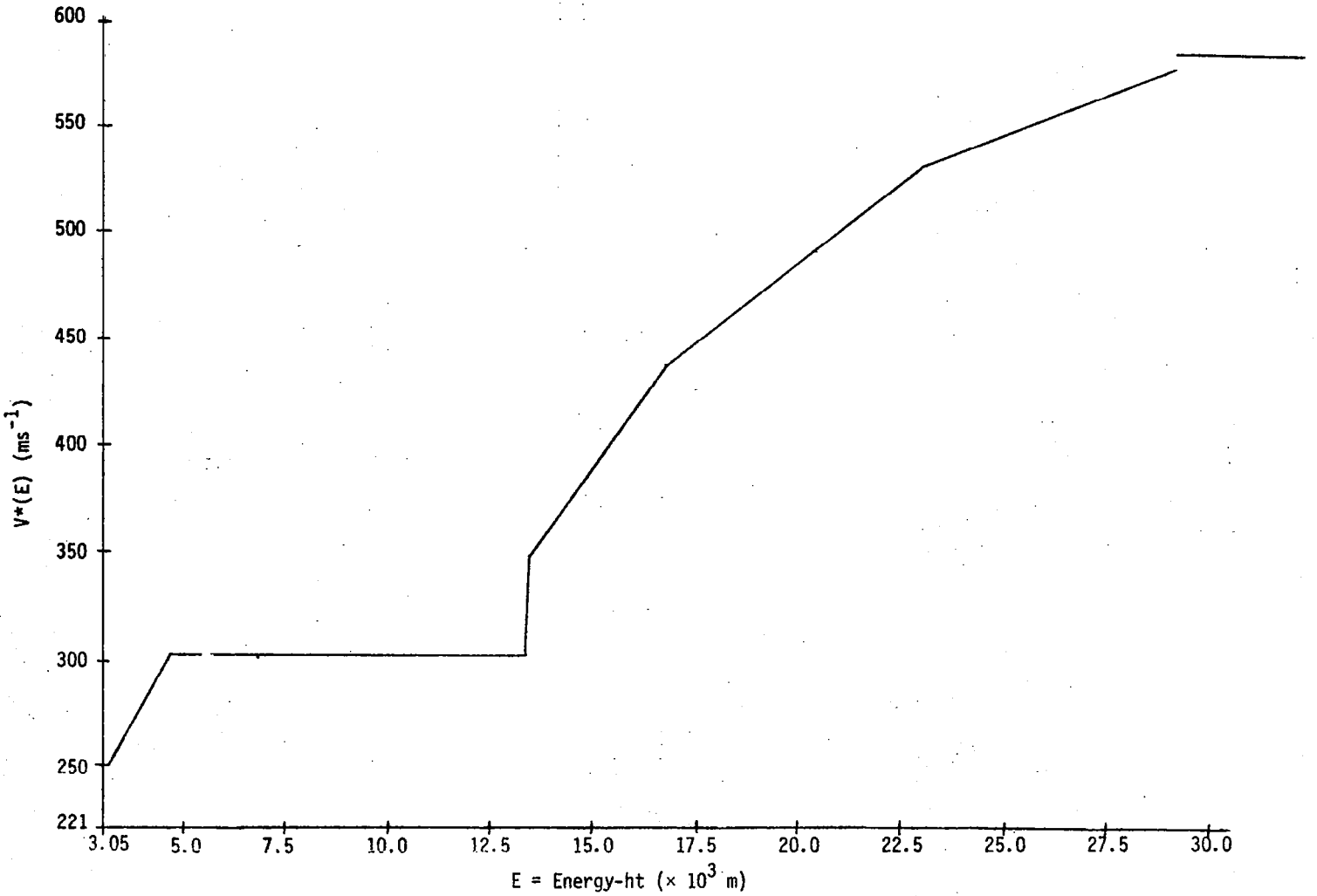


Figure 6.2.1 Piecewise Linear Approximation of V*(E)

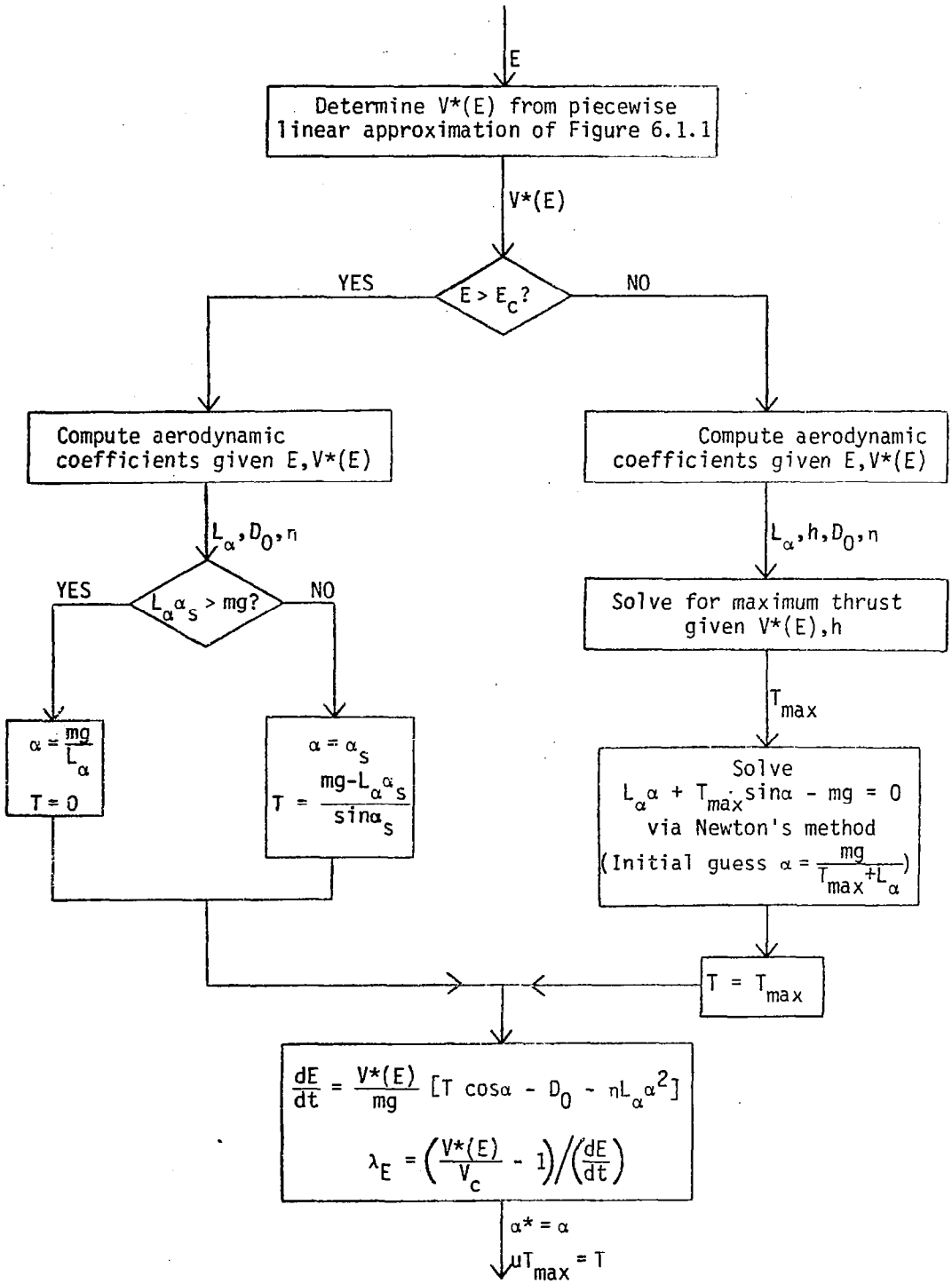


Figure 6.2.2
Flowchart of the E-Boundary Layer Computations

or estimated from sensors.

A simulation of the aircraft trajectory has been run using the controls calculated for the E-boundary layer. The simulation consisted of a predictor-corrector (Adams-Bashforth, second order) solution to equation (5.3.1), with $\alpha = \alpha^*$ and uT_{max} determined by the control algorithm at each step. The stepsize was fixed at .5 seconds, and the simulation was run past the point where the aircraft reached the cruise energy-height. The initial energy-height was 5482.7 m.

Figures 6.2.3 - 6.2.9 show the computed time histories of energy-height, altitude, velocity, Mach number, thrust, range and α respectively for the E-boundary layer trajectory of the aircraft.

The aircraft, with these boundary conditions, reaches cruise energy after 359 seconds. This is slightly slower than the 351 seconds taken by aircraft with the same boundary conditions (zoom changes in height and velocity are assumed instantaneous) in the energy state approximation of Section 6.3. This discrepancy is due to the cruise approximation of $V^*(E)$ by five piecewise linear functions.

Figures 6.2.3, 6.2.4, 6.2.7 and 6.2.9 show some chattering behavior on the cruise arc. This corresponds to oscillation around the cruise arc which can be eliminated by adding some anticipatory action as the cruise energy is approached or by switching to a linear multivariable control law in E, h and γ.

Aside from the slight differences in time-until-cruise, Figures 6.3.1 and 6.2.3 show energy-height time histories which are quite similar. The time histories of altitude, velocity, Mach number and range are also quite similar for the two algorithms, except at the beginning, where the real-time E-β solution requires a zoom-dive to the MTR locus. The thrust histories are slightly different, being smoother for the present (E-boundary layer) case. Angle-of-attack (α) appear to differ between the two solutions, but the effect of α is small.

We have also included in Figures 6.2.10 - 6.2.12 profiles of altitude versus Mach number, altitude versus range, and Mach number versus range. These are also similar to the corresponding plots for the energy state trajectories (Figures 6.3.12 - 6.3.14). The controls were computed in a closed loop fashion, and the control computation to real-time ratio was found to be approximately 1:30.

6.2.1.2 h-Boundary Layer Computations

In order to obtain the pseudocontrol γ^* and the control α^* in the h-boundary layer, we have implemented the successive approximation scheme suggested in

Subsection 5.6.1. Specifically, we have performed the following iteration:

$$\left.\begin{aligned}
&(2.1) \qquad \alpha_0 = 0 \\
&(2.2) \qquad \gamma_i = \cos^{-1}\left(\frac{\frac{V}{V_c}}{1 + \lambda\frac{*}{E}\frac{V}{mg}(T - D_0 - \eta L_\alpha \alpha_{i-1}^2)}\right) \\
&(2.3) \qquad \alpha_i' = \frac{mg}{L_\alpha + T} \\
&(2.4) \qquad \alpha_i = \alpha_i' - \frac{L_\alpha \alpha_i' + T \sin\alpha_i' - mg\cos\gamma_i}{L_\alpha + T\cos\alpha_i'}
\end{aligned}\right\} \quad i = 1,10$$

We take $\gamma^* = \gamma_{10}$ and $\alpha^* = \alpha_{10}$. Equation (2.2) is a complete form of (5.6.8), while (2.3) and (2.4) consist of a Newton iteration to solve for α in (5.6.2). If $h > h^*(E)$ we negate γ^* in order to achieve a dive.

Figure 6.2.13 shows a plot if $h^*(E)$ versus E. This was obtained from the E boundary layer simulation of this subsection. We have also plotted, in Figure 6.2.14, γ^* as a function of $h - h^*(E)$ at two energy levels: $E = 5482.7$ m and $E = 13258.9$ m. Figures 6.2.13 and 6.2.14 represent the feedback control laws for the E and γ boundary layers, respectively.

6.2.1.3 Addition of a Predictive Feedback Term

It was found that the system behaved better numerically when the nonlinear feedback terms described above were supplemented by a predictive term $K_h(h^* - h)$. The control law is $\tan\gamma^* = \tan\gamma_{10} + K_h(h^* - h)$ and α^* satisfies (5.6.2). Addition of this predictive term has the same effect as imposing a penalty cost in addition to the other cost criteria to penalize h when it is far from the "optimal" value of h computed as a control in the E-layer. Best system performance was obtained with $K_h = 0.005$.

6.2.1.4 γ-Boundary Layer Computations

To obtain α^* for the γ-boundary layer which is the lowest level of the hierarchical feedback structure, we added a predictive feedback term of the type $K_\alpha(\gamma^* - \gamma)$ subject to saturation limits of 0° and 12°. The use of SPT control together with predictive feedback terms stabilizes the control scheme and

alleviates inaccuracy due to numerical error in the control computation. Best performance was obtained for $K_\alpha = 0.6/\cos\sigma$ ($\sigma = 76°$ while turning).

6.2.1.5 The Initial Turn

In Section 5.5 where the energy-state solution was obtained, transitions in altitude and velocity were instantaneous at constant energy by the very nature of the approximation. Hence it was cost-effective to zoom to the max-turn locus, complete the turn on the locus and zoom back to the optimal energy climb path. With h and γ dynamics taken into account it is no longer clear whether this is the optimum strategy. Hence, a simulation of the aircraft trajectory was run using the controls calculated for the γ-layer and two turning strategies were implemented and compared: (a) turn on max-turn locus and (b) turn at initial states. Table 6.2.1 presents these results together with simulation results for the flight path recommended by flight manuals. Also included in this table are the results of control laws using different gains for the predictive feedback terms, as well as the energy state solution. The results in the table indicate that the total time to the cruise arc is fairly insensitive to variations in these gains (H_{GAIN},γ_{GAIN}). The energy-state approximation assumes instantaneous maneuverability in h and γ states and therefore provides a lower bound to the optimal solution. The best control strategy was found to be (b) above (Case 9 in Table 6.2.1) and was only 4% above this lower bound while current practice (Case 10) was 21% above the lower bound and 17% above the best SPT solution (b). The strategy (a) (Case 8) which involves a climb to the max-turn locus for turning was 9% greater than (b). Hence although turning is faster on the locus (2.2 sec less), the initial distance from the locus (about 3000 m altitude here) makes (a) less efficient than (b).

Figures 6.2.15, 6.2.16, 6.2.17, 6.2.18 and 6.2.19 give a view of the first 20 seconds of the trajectory ((b) above), 10 seconds of which were taken up in turning. The complete climb to cruise arc is shown in Figures 6.2.20 - 6.2.30 with the E-layer and h-layer trajectories included wherever they provide a good comparison with the γ-layer to indicate the purpose of adding boundary layers. In particular, Figure 6.2.21 shows the zoom dive in altitude (h*) demanded by the E-layer and the well-rounded descent in altitude achieved by the γ-layer. Figure 6.2.30, which is an altitude-Mach profile of the trajectory, includes in addition to the E-reduced order solution, the current practice trajectory which

is recommended by flight manuals.

6.2.2 Cruise Arc

Upon reaching the end of the energy-climb path of Section 6.2.1, some minor adjustments in altitude, velocity and flight path angle are necessary to attain the cruise values specified in Subsection 5.2.1. As mentioned in Subsection 6.2.1.1, there is a jump in velocity from 585.2 to 590.2 m/sec at $E = E_c$. In the example simulation, γ is slightly positive (4.4°) upon reaching E_c and therefore must be reduced to 0°. In order to achieve cruise values for all the states simultaneously, a backward integration was performed beginning at cruise values maintaining $\dot{E} = 0$ and α as large as possible so as to increase γ: these controls are $u = 1$ and α for $\dot{E} = 0$. In the real-time case, this integration could be done off-line and stored as a velocity-flight path angle profile. Forward integration was begun with the nominal cruise values ($V = 585.2$ m/sec, $\gamma = 4.4°$, etc.) using $\alpha = 0$ to decrease γ and u for $\dot{E} = 0$ and the states monitored to determine the intersection point with the backward trajectory to yield the optimum switching point. At this point the backward simulation controls are implemented: $u = 1$ and α for $\dot{E} = 0$ to arrive at the exact cruise values. In the example simulation, it took 17.2 secs to achieve these exact values. The same range could have been covered in 17.0 secs at $V = 590.2$ m/sec and $\gamma = 0°$, hence the cost of these corrections is very small, i.e., 0.2 secs.

6.2.3 Descent from Cruise Arc to the Terminal Aircraft States

This portion of the trajectory is a short range (30 - 40 Km) optimization problem. It has been shown in Section 4.4 that the SPT approximation breaks down as the terminal states are approached. We therefore look to the energy-state approximation of Section 5.5 to provide clues as to the optimum path to the interception point. This approximation neglects $\dot{h}$ and $\dot{\gamma}$ dynamics and so allows free (instantaneous) zoom dives and climbs. Hence, we find that the energy-state reduced order solution requires descent from the cruise arc along the max-velocity constraint until the final energy is reached after which there is a zoom climb (constant energy) to the required terminal altitude. In contrast to this solution we also have what is recommended in operations manuals for current supersonic aircraft: descent from the cruise arc along the max-velocity constraint until the final altitude is reached after which level (constant

altitude) flight is suggested to the required final energy (or velocity).

Simulation of the trajectory along the max-velocity constraint was achieved by obtaining the two controls: alpha (α) and throttle (u) so as to satisfy

$$\dot{V} = V'_{max}(E)\dot{E} \tag{2.5}$$

where $V_{max}(E)$ is as in Figure A5.1.2 and $V'_{max}(E)$ is the derivative of V_{max} with respect to E.

We now consider a simulation of the zoom climb at constant energy from the max-velocity constraint to the final altitude as required by the energy-state reduced order solution. A period of pull-out from descent on the max-velocity constraint to ascent on the constant energy (E_f) contour is necessary. Basically this involves changing the flight path angle from negative to positive as quickly as possible. Since at the final altitude, $\gamma_f = 0°$, a switching strategy exists so that on the E_f contour, a maximum γ is just reached after which the controls are switched so as to decrease γ as fast as possible ($\alpha = 0°$) to satisfy the terminal constraint on γ. There is also an optimum energy at which to begin pulling out of the V_{max} descent so as to arrive onto the E_f contour with a nonnegative flight path angle.

With $\alpha = 0$ on the energy contour (E_f), it can be easily shown that

$$h = -E_f \tan^2\gamma + h_f \sec^2\gamma \tag{2.6}$$

Hence, the switching height h_{SW} is given by

$$h_{SW} = -E_f \tan^2\gamma_{max} + h_f \sec^2\gamma_{max} \tag{2.7}$$

Hence h_{SW} is the altitude corresponding to γ_{max}; therefore a comparison of current altitude (h) to h_{SW} indicates whether γ_{max} has been reached. When $h = h_{SW}$, α must be set to zero so as to just bring γ to zero at h_f. The real-time algorithm is outlined below:

(i) descend on V_{max} constraint until pull-out energy with α and u from (2.5)

(ii) pull-out with $\alpha = \alpha_s$, $u = 1$ until $E = E_f$

(iii) maintain $\dot{E} = 0$ and allow γ to increase until $\gamma = \gamma_{max}$ determined by

testing current altitude against h_{SW}

(iv) set $\alpha = 0$ and u to maintain $\dot{E} = 0$ so that h_f is attained at constant energy (E_f) with $\gamma = 0$ at h_f

(v) complete final turn with $\dot{h} = 0$, $\dot{\gamma} = 0$, $\dot{E} = 0$ maximizing $\dot{\beta}$.

Example simulations were conducted using the above algorithm to descend from cruise to the final states. Two examples were implemented:

(1) $E_f = 1.2802 \times 10^4$ m, $h_f = 9.144 \times 10^3$ m, $\gamma_f = 0°$, $\beta_f = 90°$, $M_f = 0.89$

(2) $E_f = 1.0668 \times 10^4$ m, $h_f = 9.144 \times 10^3$ m, $\gamma_f = 0°$, $\beta_f = 90°$, $M_f = 0.56$

Note that horizontal plane constraints (i.e., x_f, y_f) are satisfied by choosing the correct cruise heading (β_c) and range on cruise (R_c). The results for (1) are presented in Table 6.2.2 which also contains the energy-state reduced order solution and an implementation of current practice. The effect of earlier pull-out from descent on the V_{max} constraint is shown by the superiority of Case 4 over Case 3 in Table 6.2.2. Note that as before the energy-state approximation solution provides a lower bound to the optimal solution since instantaneous maneuverability is assumed in the h and γ states. The best real-time solution (Case 4) was 24% above this lower bound while current practice was 28% above the lower bound and about 4% above Case 4.

The closeness of the result of Case 2 (current practice) to the best real-time solution (Case 4) motivated the simulation of example (2) above, the results of which appear in Table 6.2.3. Again the energy-state reduced order and current practice solutions are included. Here the best real-time solution (Case 4) is about 19% above the lower bound while current practice (Case 2) is 65% above the lower bound and about 38% above the best real-time solution. Hence taking the results of (1) and (2) together, current practice is almost as good as our best real-time solution in (1) where the final Mach number is high ($M_f = 0.89$) while in example (2) current practice is considerably less satisfactory than the best real-time solution. In (2) the final Mach number required is lower ($M_f = 0.56$) and we see that level flight through lower Mach number regions is time-consuming. Hence the optimum strategy for descent from the cruise arc is dependent upon the particular final states required of the airplane.

The whole trajectory of example (2) above, i.e., Case 4 of Table 6.2.3, is displayed in Figures 6.2.31 - 6.2.43, of which Figures 6.2.39 - 6.2.41 represent the three control (α, u, σ) time-histories. We have added in the E reduced-order

solution and current practice wherever the comparisons are meaningful. Note that for the purpose of comparing the different cases in Table 6.2.3 a common range (68.86 Km) had to be considered so Case 4 includes extra range on cruise. With the exception of Figures 6.2.32, 6.2.42 and 6.2.43, this extra range is not shown in the figures. Common range considerations were also made in Tables 6.2.1 and 6.2.2 with extra range added onto cruise wherever necessary.

6.3 Numerical Results Using a Real-Time Algorithm Based on the Energy State Approximation

A real-time trajectory was obtained using the algorithm described in Subsection 5.5.2 with turning only on the max-turn locus. Initial and final conditions chosen were:

$h_0 = 3.353 \times 10^3$ m $\qquad h_f = 8.230 \times 10^3$ m

$M_0 = 0.6$

$E_0 = 5.517 \times 10^3$ m $\qquad E_f = 1.158 \times 10^4$ m

$x_0 = 0$ $\qquad x_f = 212.0$ Km

$y_0 = 0$ $\qquad y_f = 169.6$ Km

$\beta_0 = 30°$ $\qquad \beta_f = 120°$

Figure 6.3.1 gives the energy changes with time along the trajectory. Roughly two thirds of the trajectory consists of reaching the cruise arc while the section from the cruise to the final conditions is by comparison much shorter: the rate of energy change being higher along the bank angle chatter arc.

Figures 6.3.2 to 6.3.4 reveal the zoom dives and climbs which reflect the instantaneous changes in altitude and velocity allowed by the energy-state approximation. The real-time approximation used here which allows turning only on the max-turn locus is clearly visible in Figure 6.3.5 which shows how the required 90° turn is broken up into an 8.3° initial turn and 81.7° final turn. Figure 6.3.6 shows the many variations in bank angle along the trajectory. By

the approximation mentioned above turning is with $\sigma = \sigma_m$ and $\sigma = 0$ for straight flight while $\sigma = \pm\min(\sigma_m, \sigma_s)$ during chattering to maintain average $\beta = 0$. The thrust control history is quite simple: $T = T_{max}$ all the way to the cruise arc, $T = T_c < T_{max}$ while cruising since the cruise arc is an intermediate thrust singular arc followed by $T = 0$ while the bank angle chatters and then $T = T_{max}$ for the final turn.

The horizontal plane projections of the trajectory appear in Figures 6.3.8 - 6.3.11. Figure 6.3.11 indicates that a very small portion of the trajectory time is spent in turning on the max-turn locus as compared to the straight-flight portions. The alpha control history is given in Figure 6.3.12. The altitude-Mach number profile of Figure 6.3.13 provides a very interesting view of the trajectory. From the initial conditions which are on the max-turn locus, the trajectory executes the initial turn followed by a zoom dive to the minimum time energy climb path which it follows to the cruise point. After cruising the bank angle is allowed to chatter while moving down the max velocity constraint. The trajectory leaves the constraint by a zoom climb to the max-turn locus where it executes the final turn to reach the end conditions which are also on the locus. If the initial or final conditions were not on the locus, the trajectory would include zoom dives or climbs to reach the locus. Figure 6.3.14 is an altitude-range profile where the rapid changes in altitude are clearly visible while Figure 6.2.15 shows the changes in Mach number with range.

In order to perform the t_f iterations of the algorithm presented in Subsection 5.5.2, some assumptions have to be made about the target dynamics. We assume that it moves in a straight line at constant altitude and velocity. Then the steps in the algorithm are as follows:

(1) obtain initial estimate of interception point by calculating the shortest distance to target path from pursuer

(2) evaluate t_f = time for target to reach this point

(3) implement (2) - (12) of algorithm in Subsection 5.5.2

(4) compute pursuer trajectory time (t_1) to interception point

(5) define the difference as $f(t_f) = t_f - t_1$ and perform approximate Newton iteration as follows:

$$t_{f_{new}} = t_{f_{old}} - \frac{f(t_{f_{old}})}{f'(t_{f_{old}})} \simeq t_1$$

(6) go to (3) if accuracy on t_f is not met.

(1) - (6) above were implemented for two cases given in Figures 6.3.16 and 6.3.17. The first case required six iterations for a 180° turn to an accuracy of 0.1 secs. The second case (Figure 6.3.17) required 11 iterations. This was the worst case: 90° turn to a target which has already passed the point where its path is at the shortest distance from pursuer. Hence the two cases indicate good convergence. Note that a better initial estimate of the interception point (step (1) above) using proportional navigation or some other scheme would reduce the number of iterations.

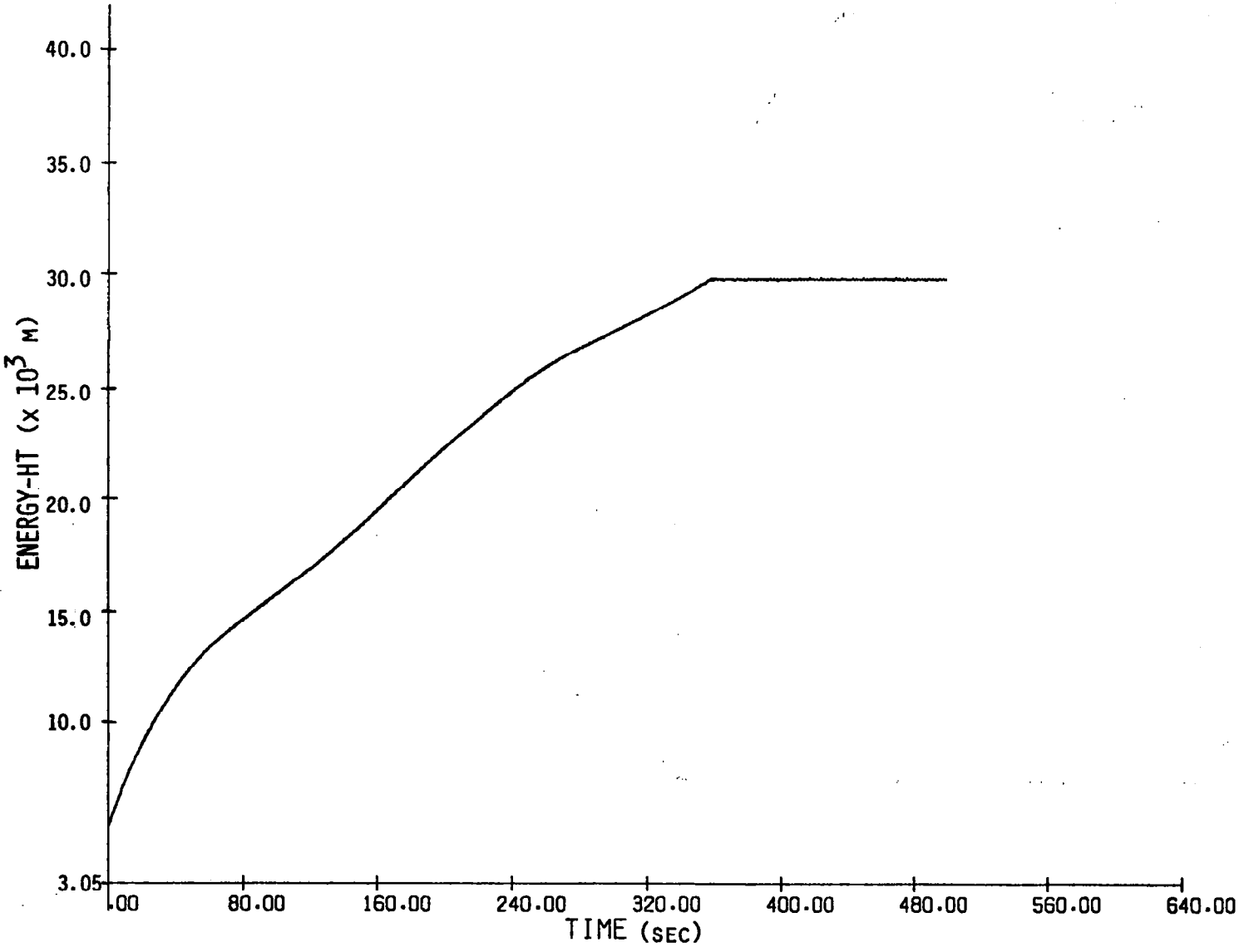


Figure 6.2.3 Time History of Energy-Height

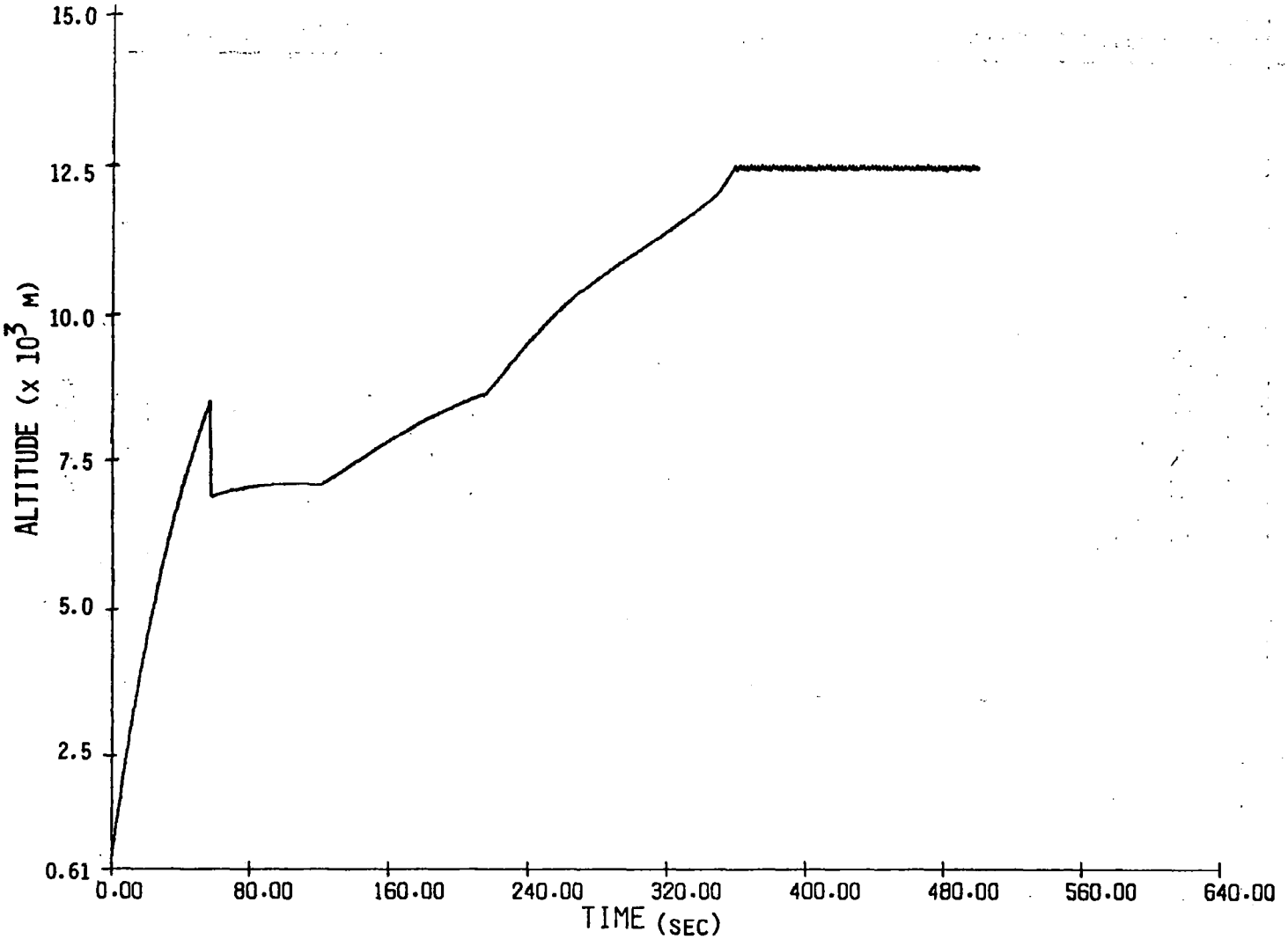


Figure 6.2.4 Time History of Altitude

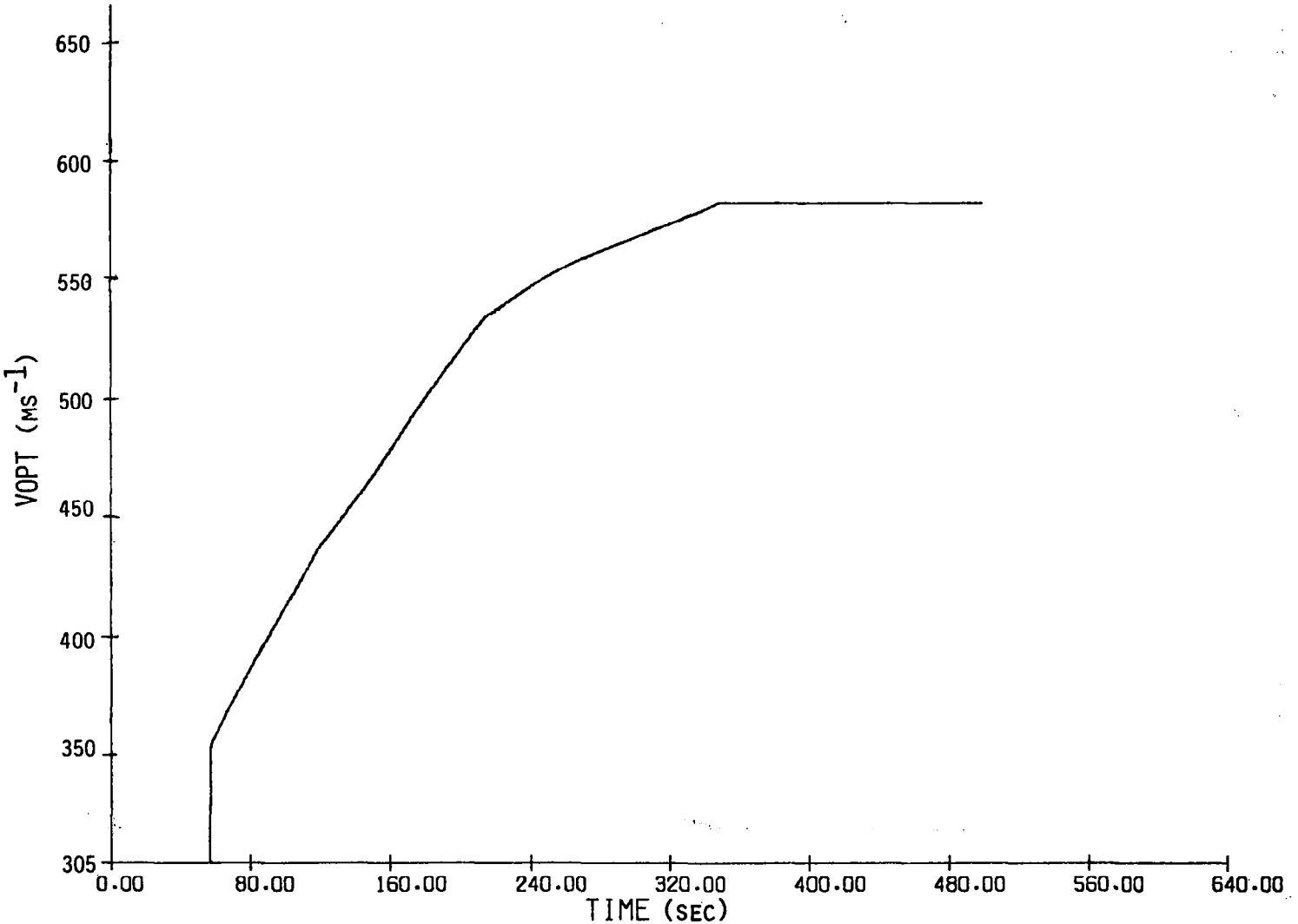


Figure 6.2.5 Time History of Velocity

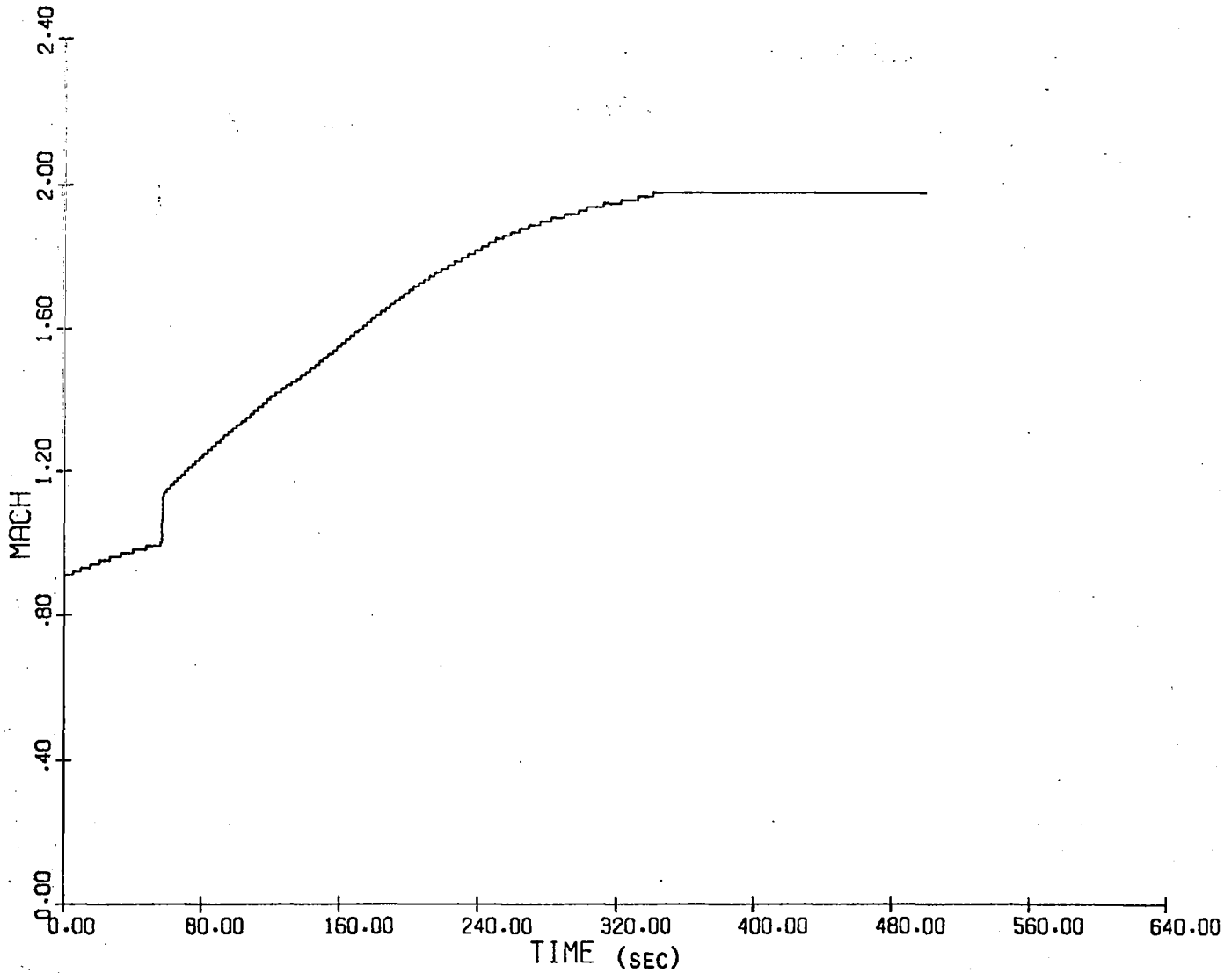


Figure 6.2.6 Time History of Mach Number

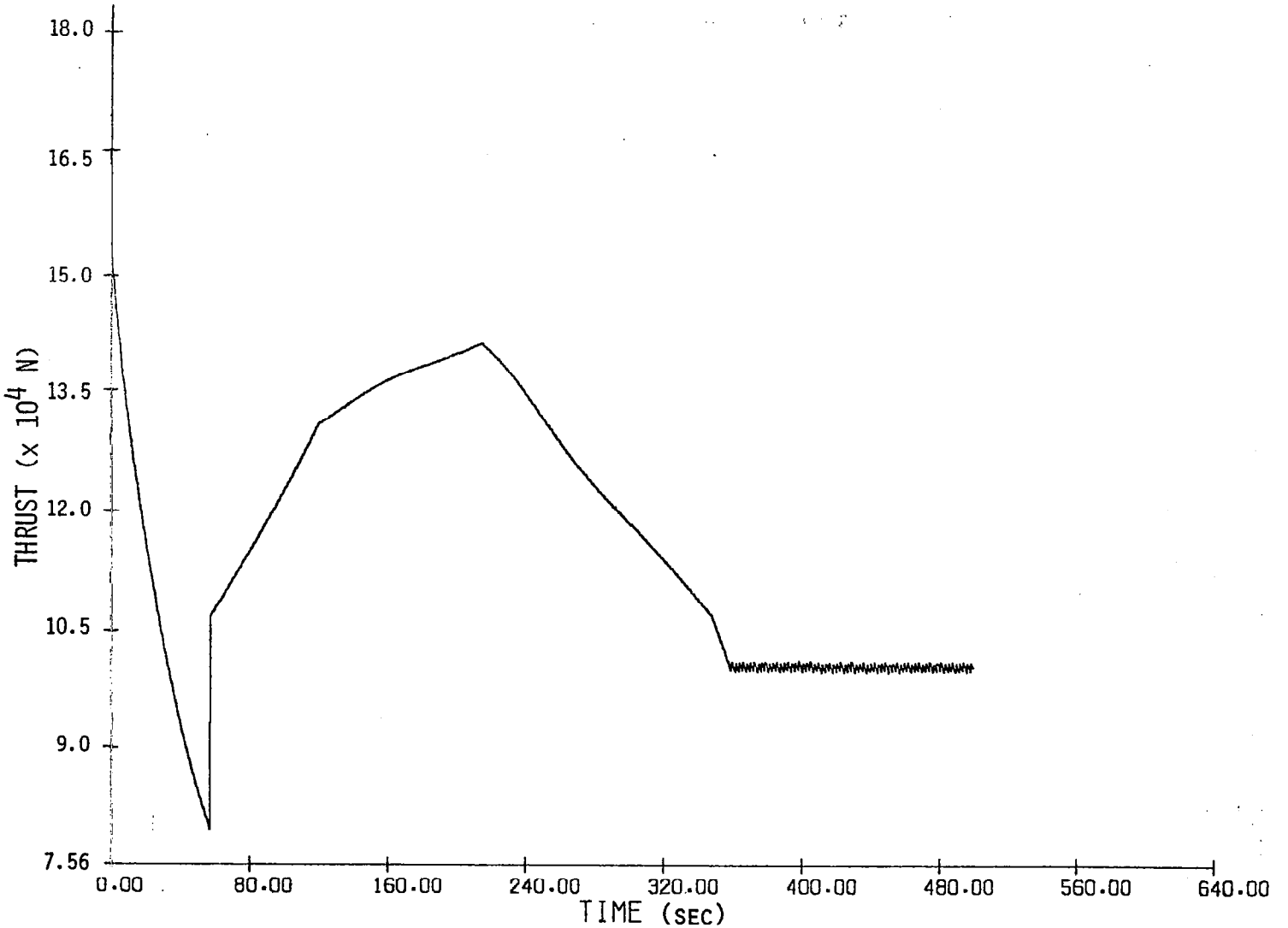


Figure 6.2.7 Time History of Thrust

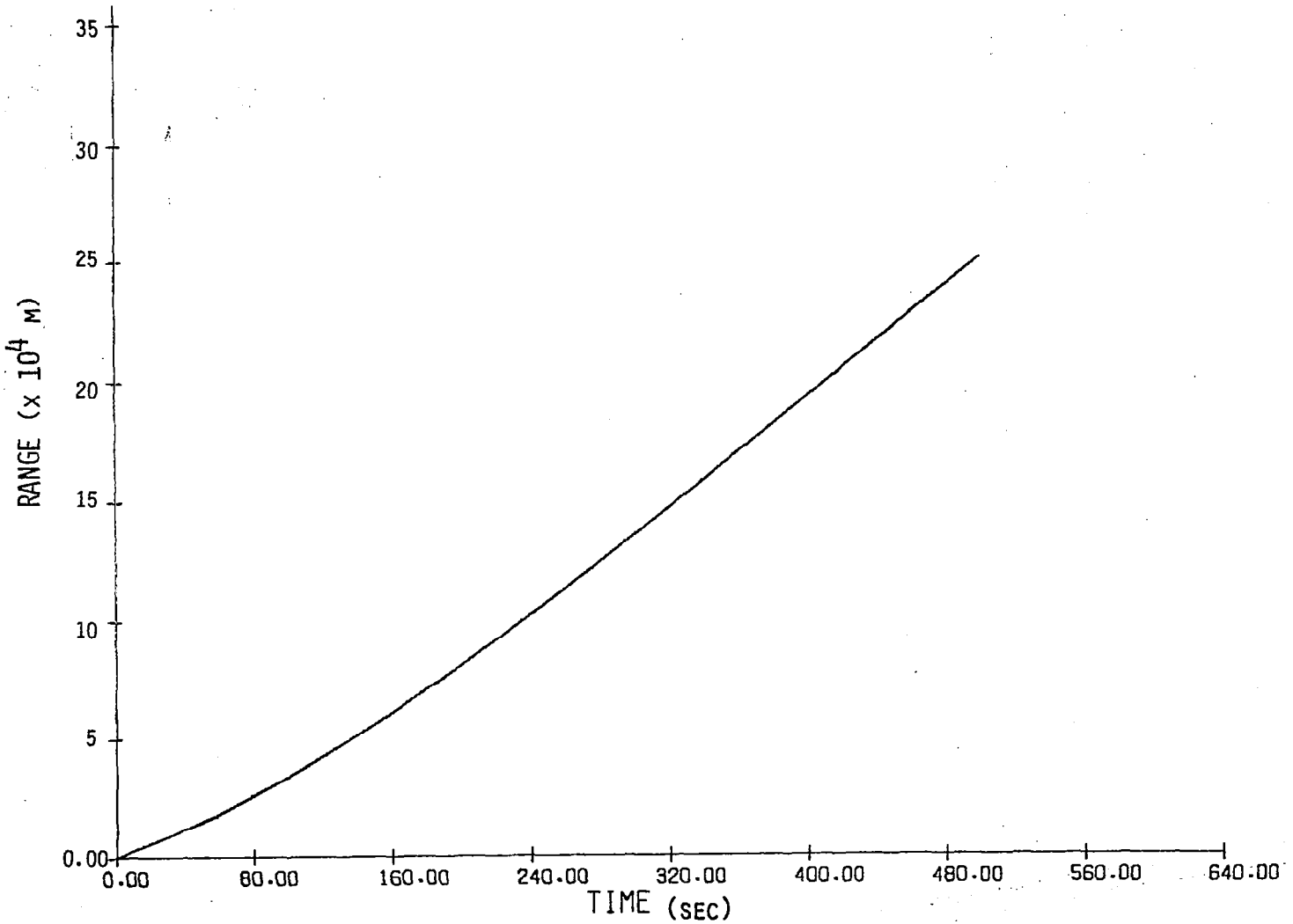


Figure 6.2.8 Range as Function of Time

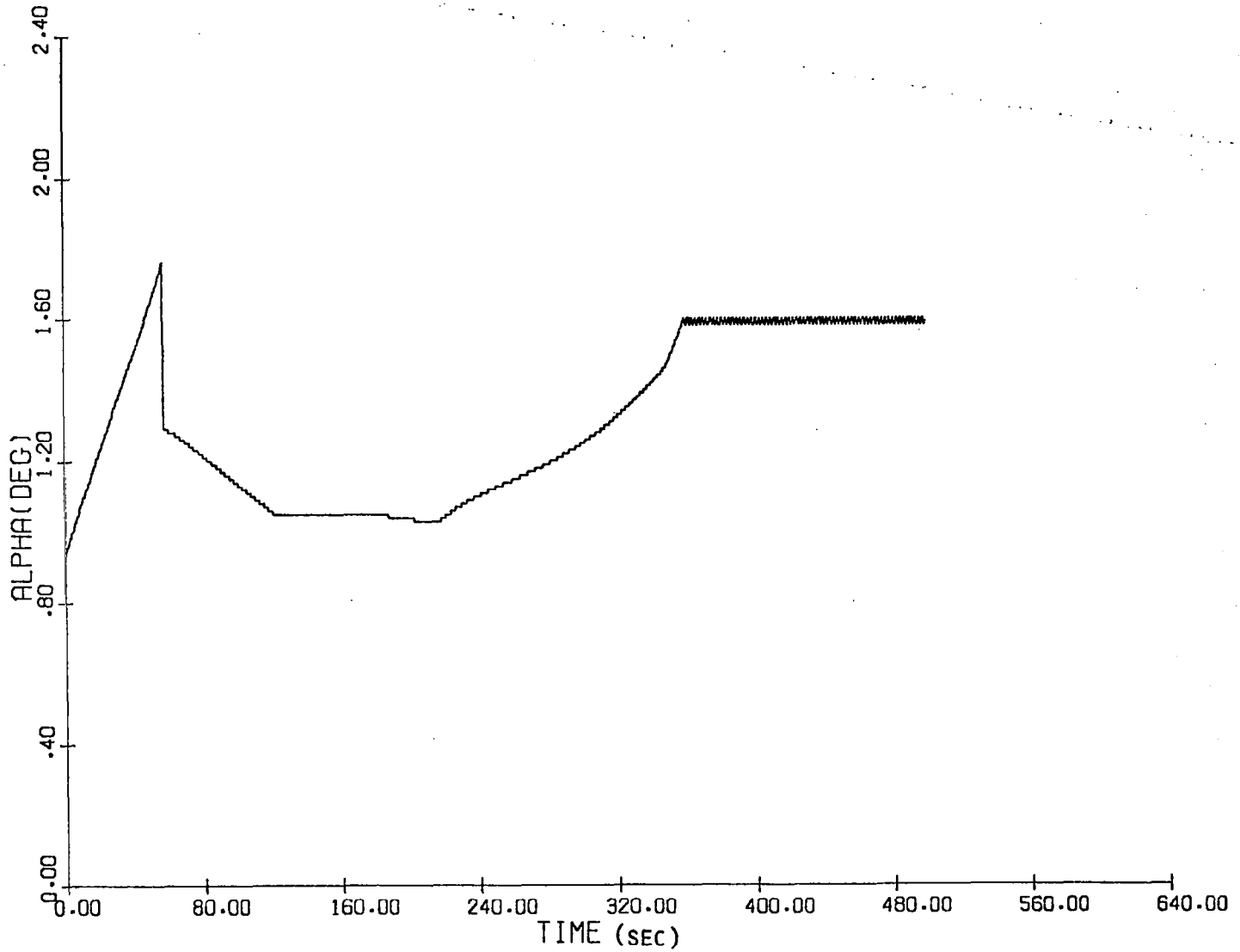


Figure 6.2.9 Time History of Angle-of-Attack

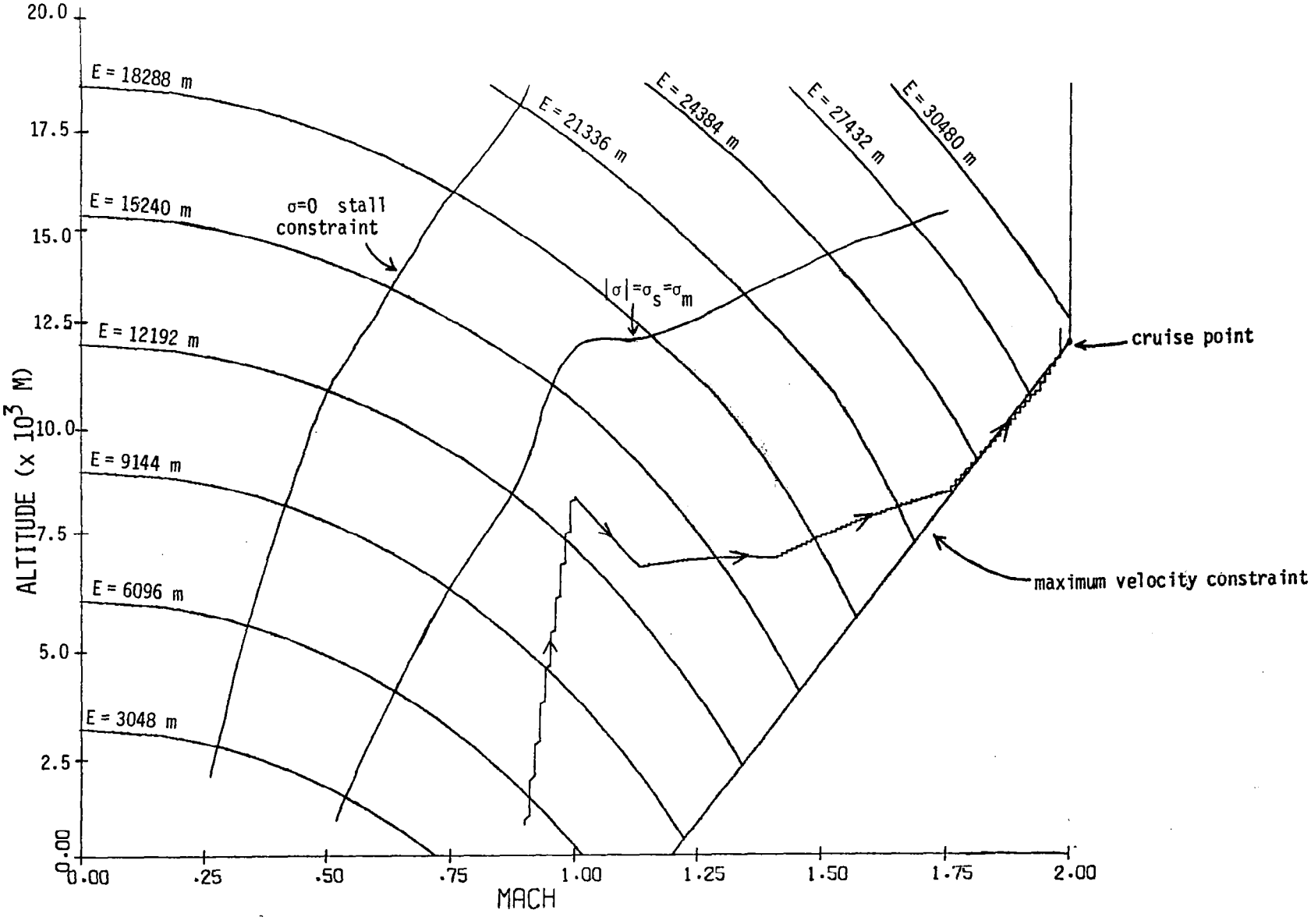


Figure 6.2.10 Altitude vs. Mach Number

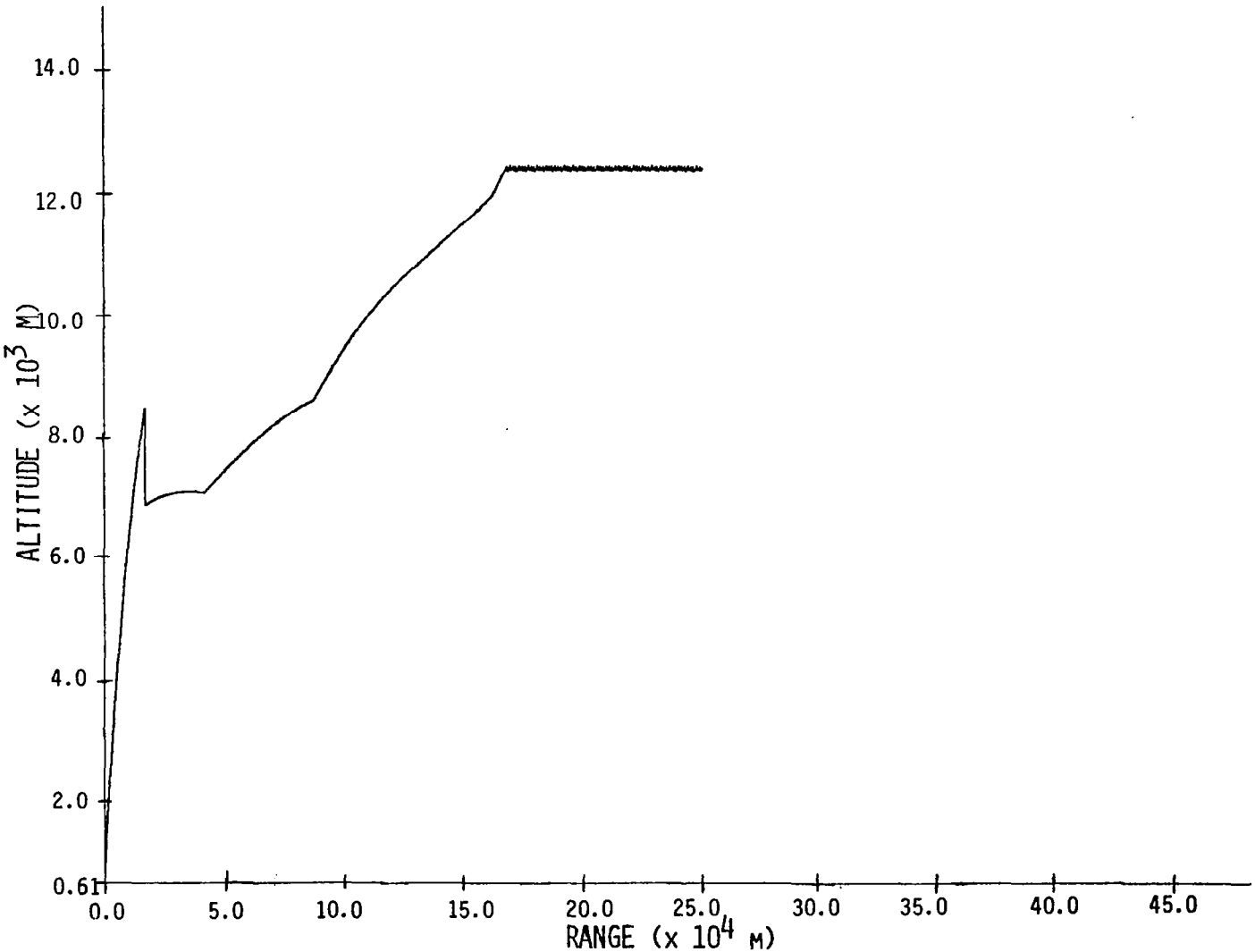


Figure 6.2.11 Altitude vs. Range

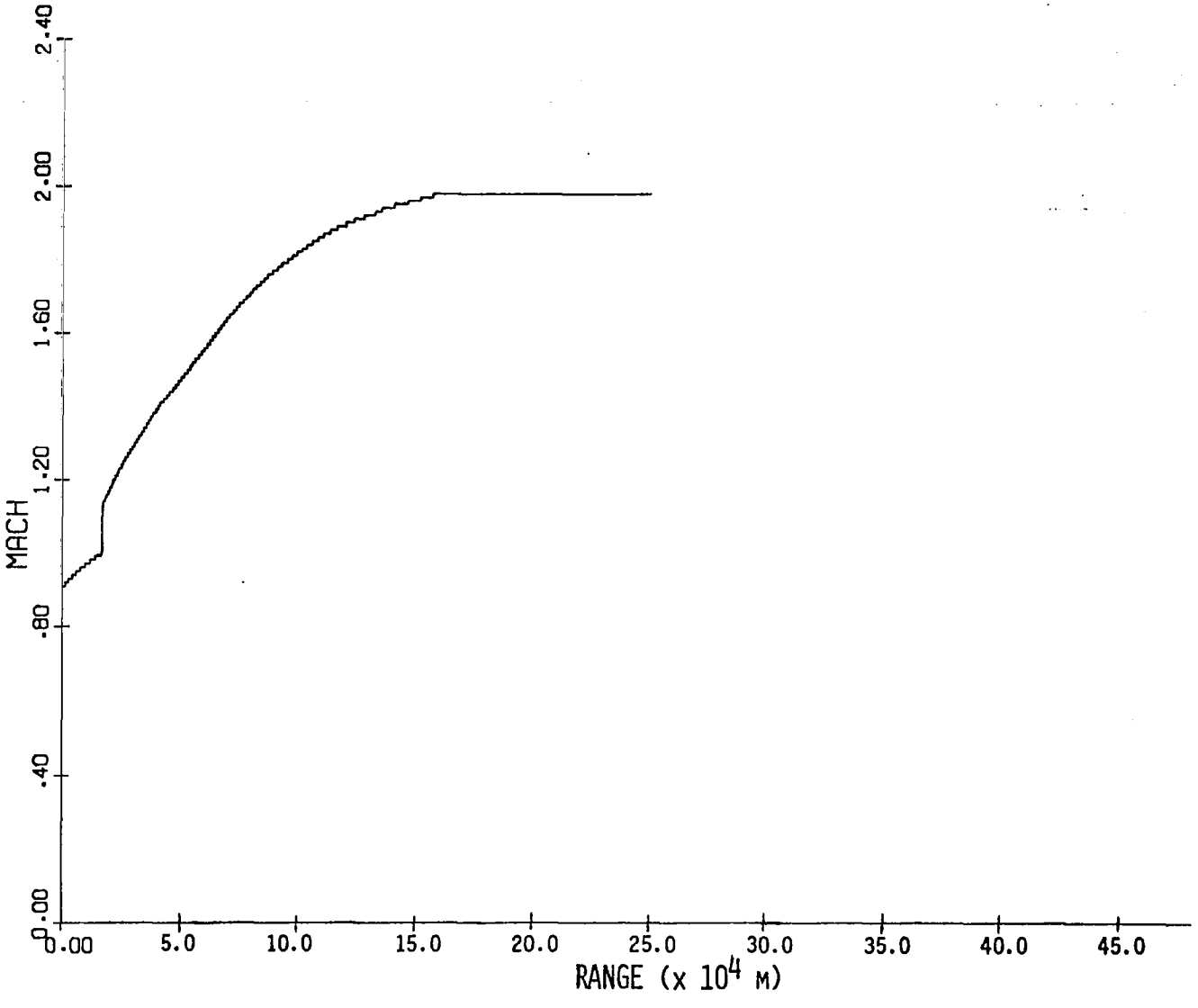


Figure 6.2.12 Mach Number vs. Range

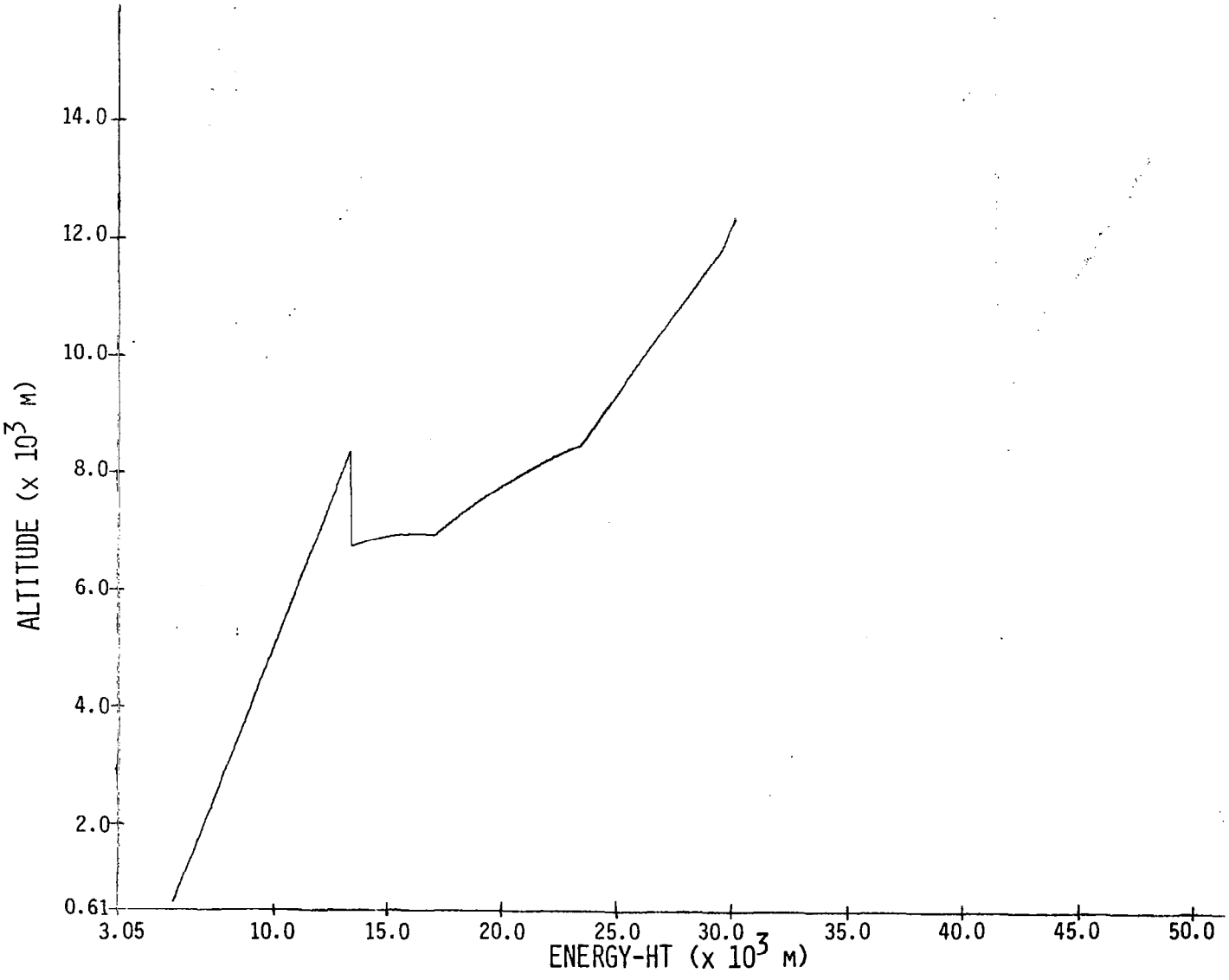
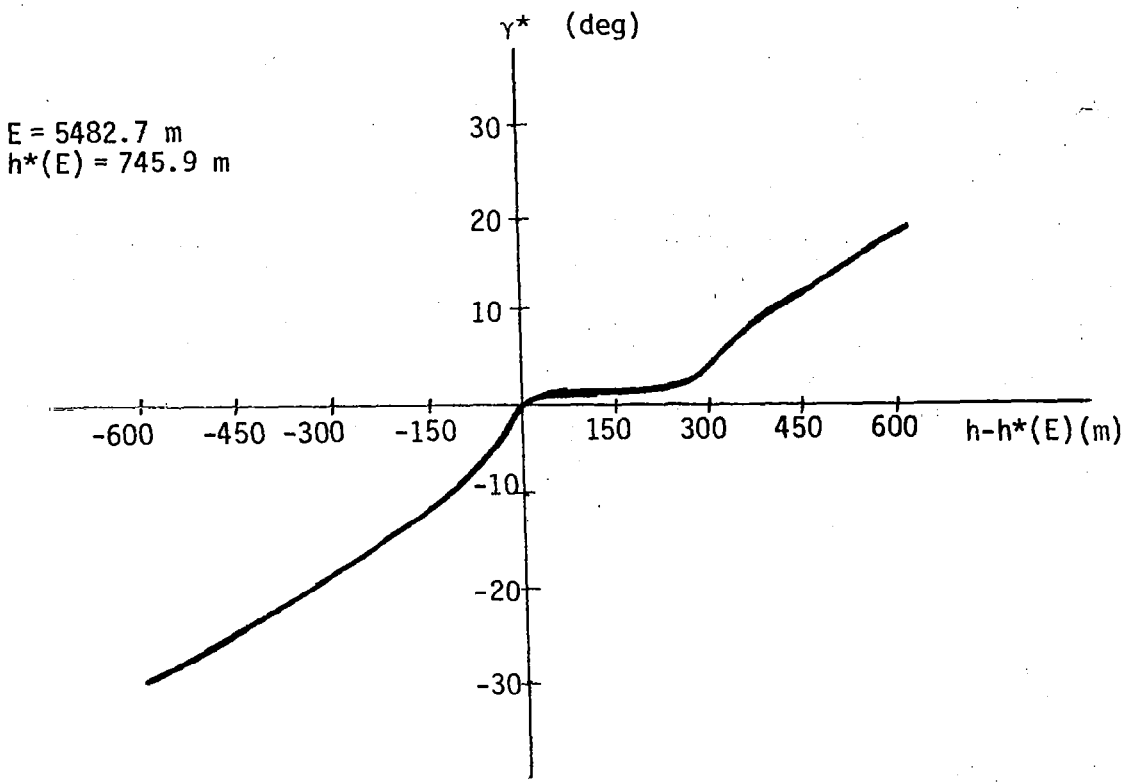


Figure 6.2.13 Feedback Control Law for the E-Boundary Layer



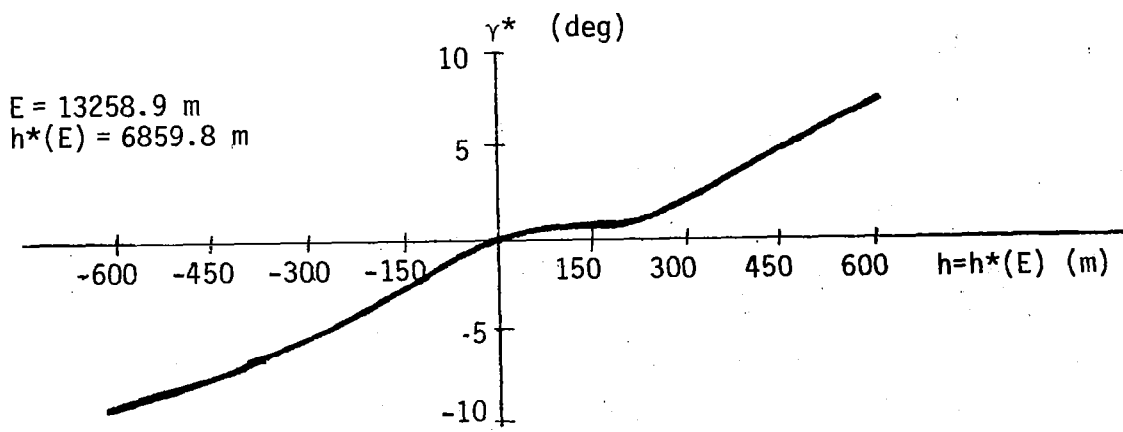


Figure 6.2.14
Feedback Control Law for the h-Boundary Layer

Table 6.2.1 Summary of Simulation Results for Trajectories to Cruise Arc

CASE	Time to cruise arc, starting at H = 593.4 m, M = 0.9, energy = 5482.7 m Final range = 182.9 Km altitude = 12192 m, Mach = 2.0, energy = 29949.3 m	REMARKS
1. Energy-layer	380.9 secs	This approximation treats h and γ as controls and holds the derivatives $\dot{h}$ and $\dot{\gamma}$ equal to 0. Intuitively, it assumes that h and γ change much more quickly than E. That is, h and γ are characterized by sudden changes (boundary layers) for short times and almost steady behavior for the rest of the time.
2. h-layer with nonlinear and linear feedback gain on tanγ - linear gain = 0.01 on (H*(E)-H).	387.1 secs	This approximation treats γ as a control, assuming that γ varies faster than h does. Linear feedback is also used. It has the same effect as imposing a quadratic penalty cost in addition to the other cost criteria to penalize h when it is far from the optimal h-control of the E-layer.
3. h-layer with $h_{GAIN} = 0.005$	386.9 secs	See remarks above.
4. γ-layer with $h_{GAIN} = 0.005$ and linear feedback gain on α, $\gamma_{GAIN} = 0.2$ $0 \leq \alpha \leq 12°$	397.2 secs	The control uses a nonlinear SPT control together with linear feedback terms which represent quadratic penalties for being away from "optimal" values of h computed as a control in the E-layer and of γ computed as a control in the h layer. This procedure stabilizes the control scheme and alleviates inaccuracy due to numerical error in the control computation.

Table 6.2.1 (cont.)

CASE	TIME TO CRUISE ARC	REMARKS
5. γ-layer with $h_{GAIN} = 0.005$ $\gamma_{GAIN} = 0.4$ $0 \leqq \alpha \leqq 12°$	397.1 secs	See remarks in 4.
6. γ-layer $h_{GAIN} = 0.01$ $\gamma_{GAIN} = 0.6$ $0 \leqq \alpha \leqq 12°$	398.1 secs	See remarks in 4.
7. γ-layer $h_{GAIN} = 0.005$ $\gamma_{GAIN} = 0.4$ $0 \leqq \alpha \leqq 12°$	397.0 secs	See remarks in 4.
8. γ-layer with 90° turn on max-turn locus $0 \leqq \alpha \leqq 12°$ $\sigma = 76°$ (on locus)	438.5 secs	Path to locus simulated using α flip-flop control and γ constrained to be less than 47°. Aircraft flies to the maximum rate turn locus to turn.
9. γ-layer with 90° turn begun at initial state $\gamma_G = 0.6/\cos\sigma$ $h_G = 0.005$ $0 \leqq \alpha \leqq 12°$ $\sigma = 76°$ while turning	403.0 secs	The aircraft executes its turn without returning to the max rate turn locus. This strategy is much more efficient than (8) given the initial distance from the locus (about 3000 m altitude).
10. Current practice: turn using $\sigma = 60°$, at constant velocity, constant altitude acc. to $M = 0.9$, constant Mach # climb to 12192 m and $\gamma = 21.8°$, constant altitude acc. to cruise point.	472.7 secs	This scheme is typical of paths recommended in operations manuals for current supersonic aircraft.

Table 6.2.1 (cont.)

CASE	TIME TO CRUISE ARC	REMARKS
11. Energy layer with zoom to max turn locus, 90° turn on the locus and zoom back to energy layer (Parsons solution)	388.7 secs	See remarks for 1. The additional 7.8 secs is the time taken for a 90° turn on the max turn locus.

Table 6.2.2 Summary of Simulation Results for Terminal Trajectories off the Cruise Arc

Case	Time from cruise arc: energy=29946.6 m, h=12192 m M=2.0 to final states: energy=12801.6 m, M=0.89, h=9144 m, γ=0°, range=46.19 Km	Remarks
1. Energy-state approximation solution: down max-velocity constraint to E_f=12801.6 m; zoom climb to h=9144 m at constant energy; 90° turn at E=12801.6 m, h=9144 m	92.5 secs	See Parsons (1972) for this solution. This is a lower bound for the optimal solution since instantaneous maneuverability is assumed in h and γ states
2. Current practice: down max-velocity constraint until h=9144 m and γ=0°; level flight at h=9144 m to E=12801 m, M=0.89; 90° turn with σ, α, u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0	118.6 secs	This scheme is typical of paths recommended in operations manuals for current supersonic aircraft.
3. Real-time solution: down max-velocity constraint till energy=12801.6 m; pull-out with $\alpha=\alpha_s$, u=1 till $\gamma=\gamma_{max}$ ($\geq$21°); $\dot{\gamma}$=0 with u=1 till energy=12801.6 m again; $\dot{E}$=0, α=0 so as to obtain γ=0 at h=9144 m; 90° turn with σ, α, u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0	130.9 secs	This solution attempts to follow (1) as closely as possible; γ and h dynamics require a period of pull-out from descent on max-velocity constraint. The final state γ_f=0° requires γ to be limited to γ_{max} while on energy contour.
4. Best real-time solution: down max-velocity constraint till energy=15849.6 m; pull-out with $\alpha=\alpha_s$, u=1 till energy=12801.6 m; maintain $\dot{E}$=0 till γ reaches γ_{max} (=40°); keep $\dot{E}$=0 with α=0 so as to reach h=9144 m with γ=0°; 90° turn with α, σ, u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0	114.6 secs	This solution is better than (3) because of earlier pull-out from descent on V_{max} constraint (3048 m sooner).

Table 6.2.3 Summary of Simulation Results for Terminal Trajectories off the Cruise Arc

Case	Time from cruise arc: Energy=29946.6 m, h=12192 m, M=2.0 to final states: Energy=10668 m M=0.56, h=9144 m, γ=0°, range=68.86 Km	Remarks
1. Energy-state approximation solution: down max-velocity constraint to E_f=10668 m; zoom climb to h=9144 m at constant energy; 90° turn at E=10668 m, h=9144 m.	138.8 secs	See Parsons (1972) for this solution. This is a lower bound for the optimal solution since instantaneous maneuverability is assumed in h and γ states.
2. Current practice: down max-velocity constraint until h=9144 m and γ=0°; level flight at h=9144 m to E=10668 m, M=0.56; 90° turn with σ,α,u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0.	228.0 secs	This scheme is typical of paths recommended in operations manuals for current supersonic aircraft.
3. Real-time solution: down max-velocity constraint till E=13716 m; pull-out with $\alpha=\alpha_s$, u=1 till E=10668 m, maintain $\dot{E}$=0 till γ reaches γ_{max} (=59.6°); keep $\dot{E}$=0 with α=0 so as to reach h=9144 m with γ=0°; 90° turn with σ,α,u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0	169.1 secs	This solution follows (1) with a pull-out from descent on V_{max} constraint 10000 ft before reaching E_f=10668 m.
4. Best real-time solution: down max-velocity constraint till E=15240 m; (earlier pull-out than in (3)); pull-out with $\alpha=\alpha_s$, u=1 till E=10668 m; maintain $\dot{E}$=0 till γ reaches γ_{max} (=59.6°); keep $\dot{E}$=0 so as to reach h=9144 m with γ=0°; 90° turn with σ,α,u so as to maintain $\dot{E}$=0, $\dot{h}$=0, $\dot{\gamma}$=0.	165.2 secs	This solution is better than (3) because of earlier pull-out (1524 m sooner than (3)) from descent on V_{max} constraint

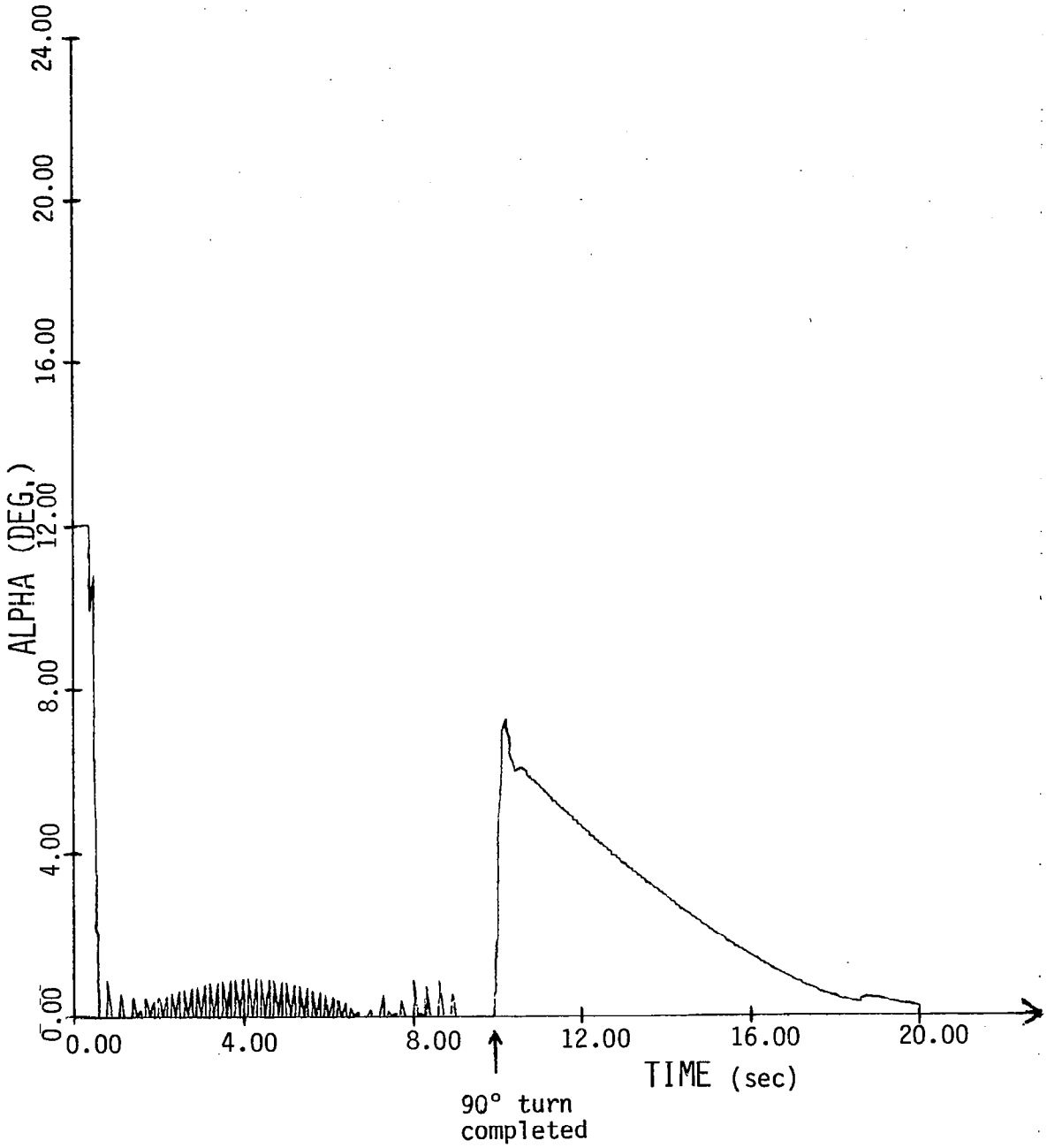


Figure 6.2.15 Alpha vs. Time While Turning

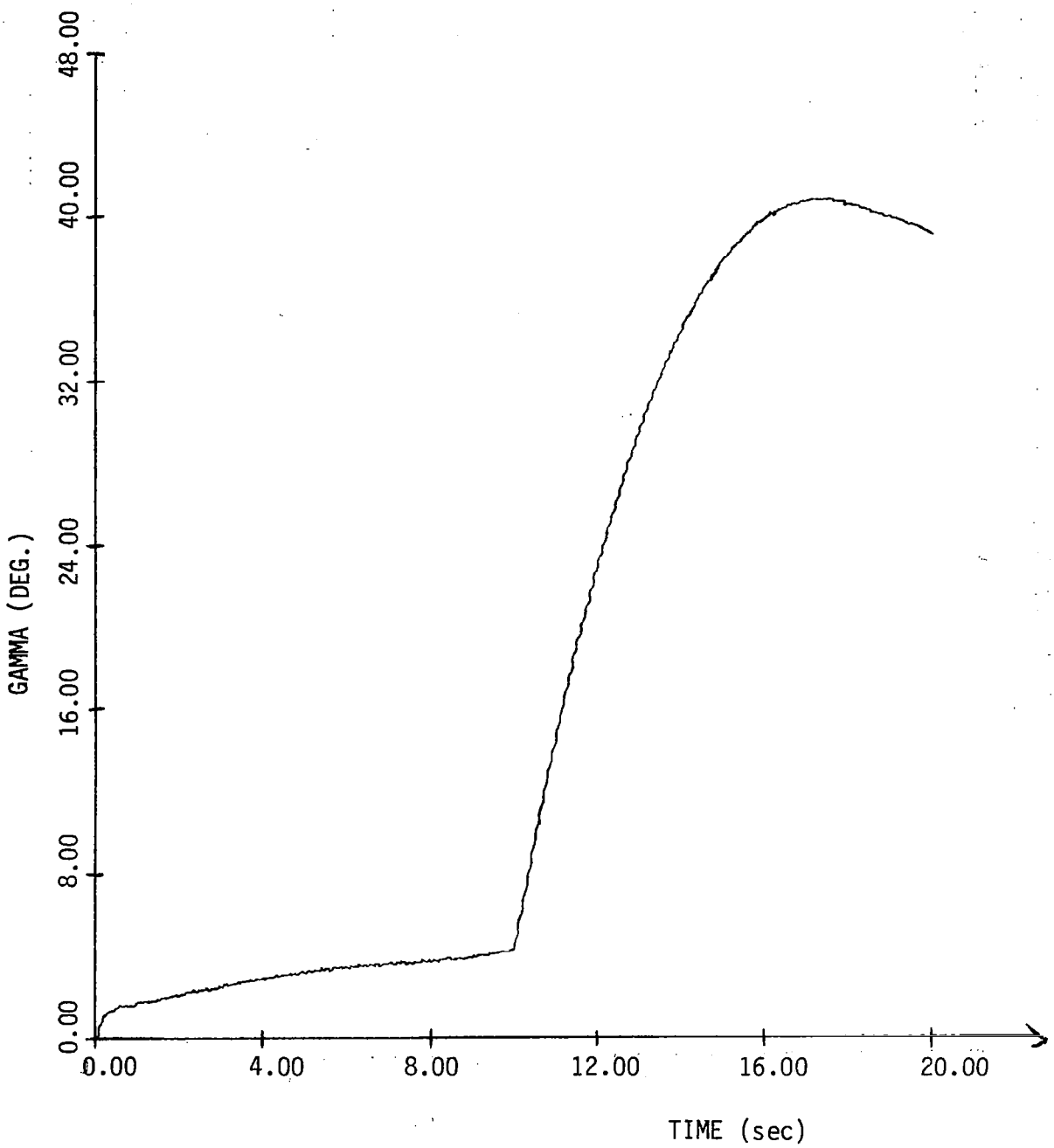


Figure 6.2.16 Variations in Gamma While Turning

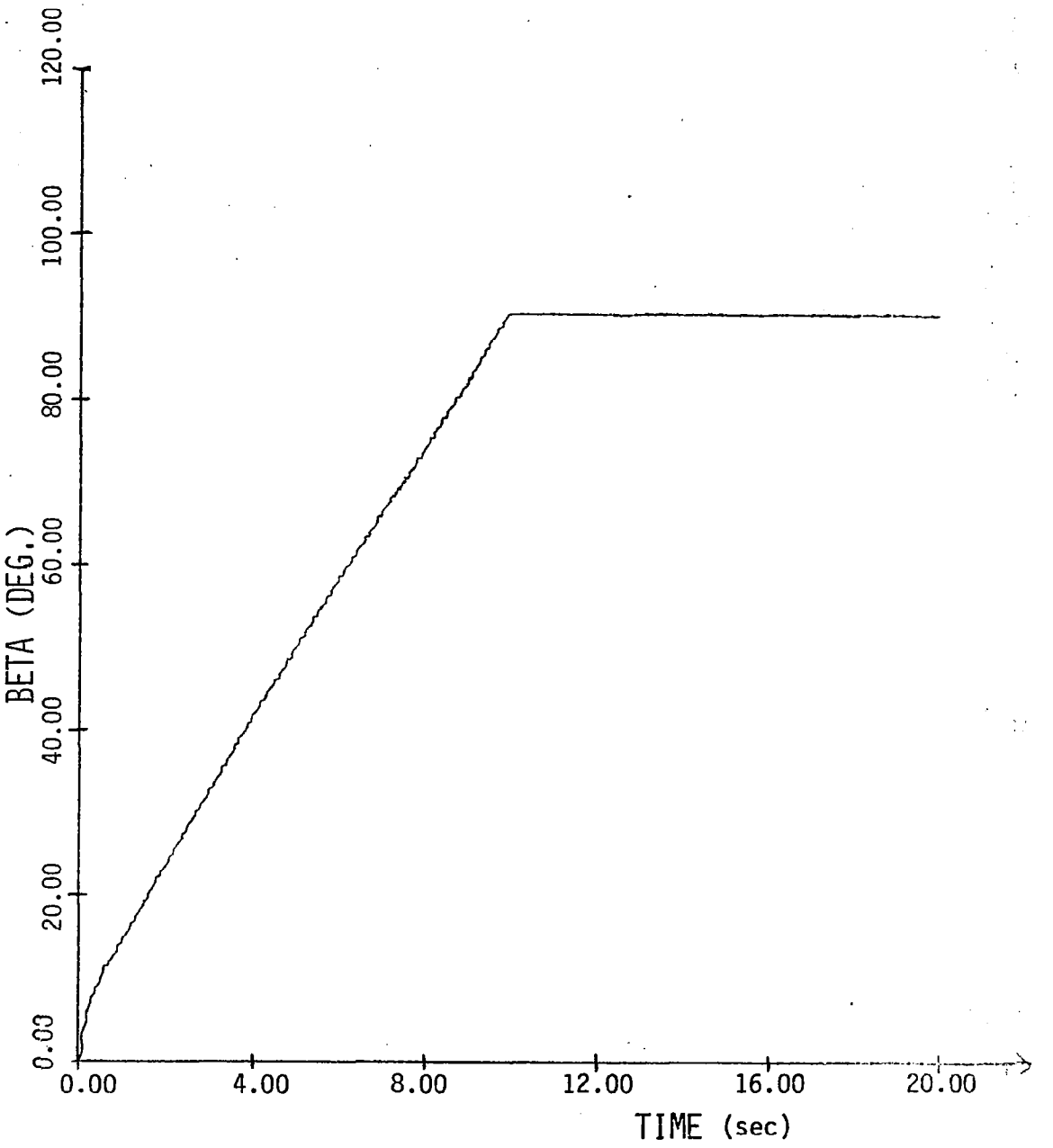


Figure 6.2.17 Changes in Beta With Time

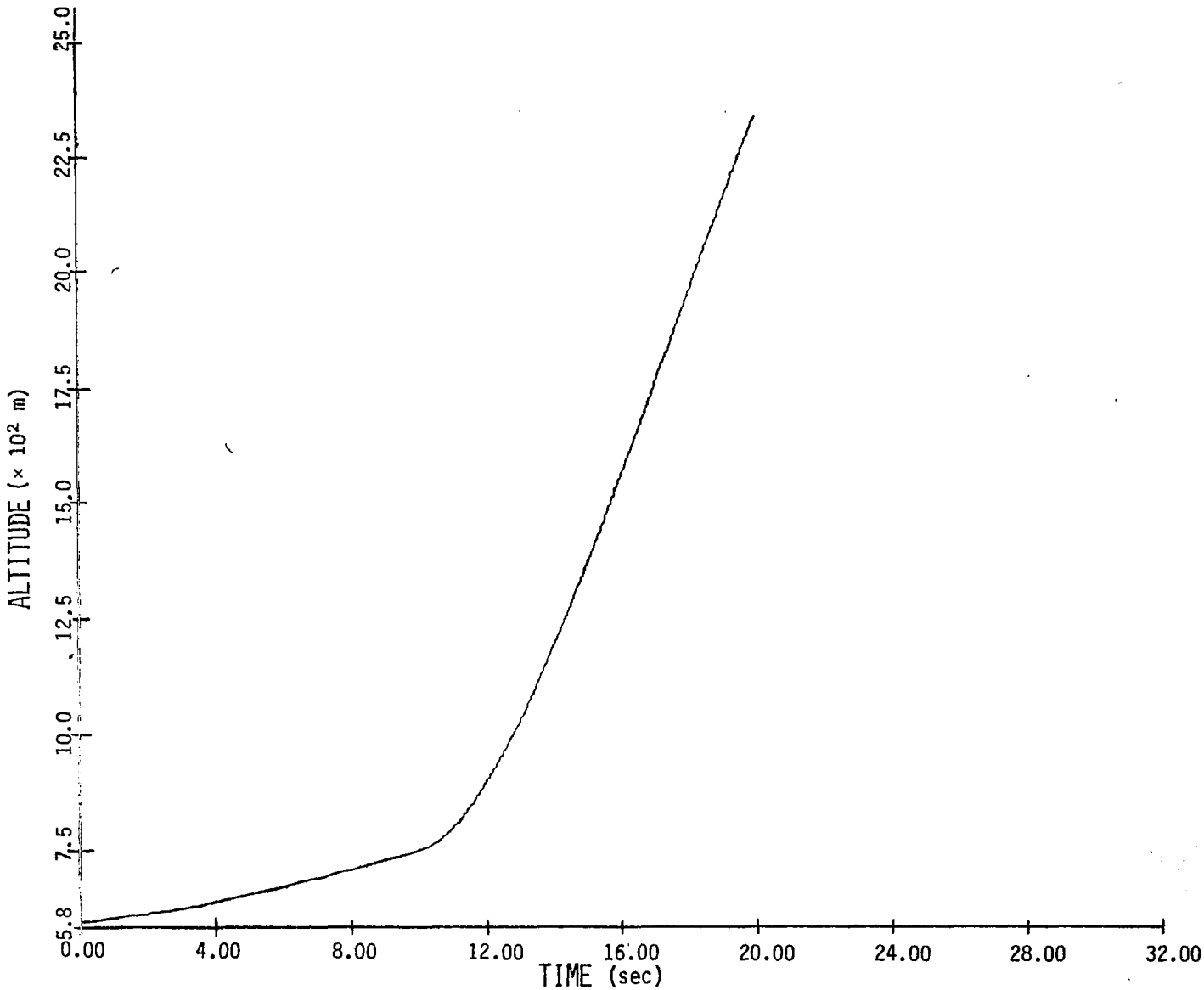


Figure 6.2.18 Altitude vs. Time While Turning

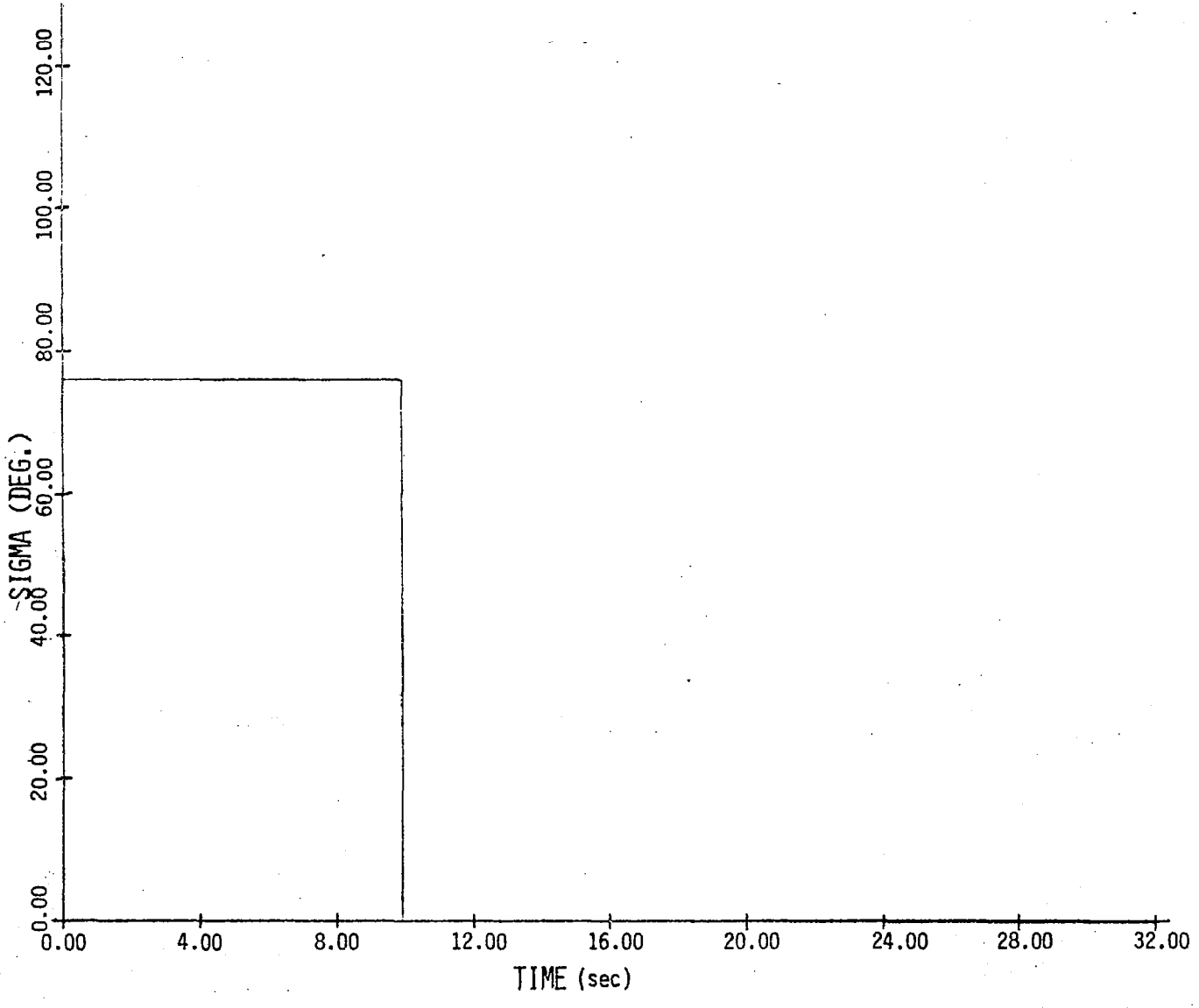


Figure 6.2.19 Sigma Control While Turning

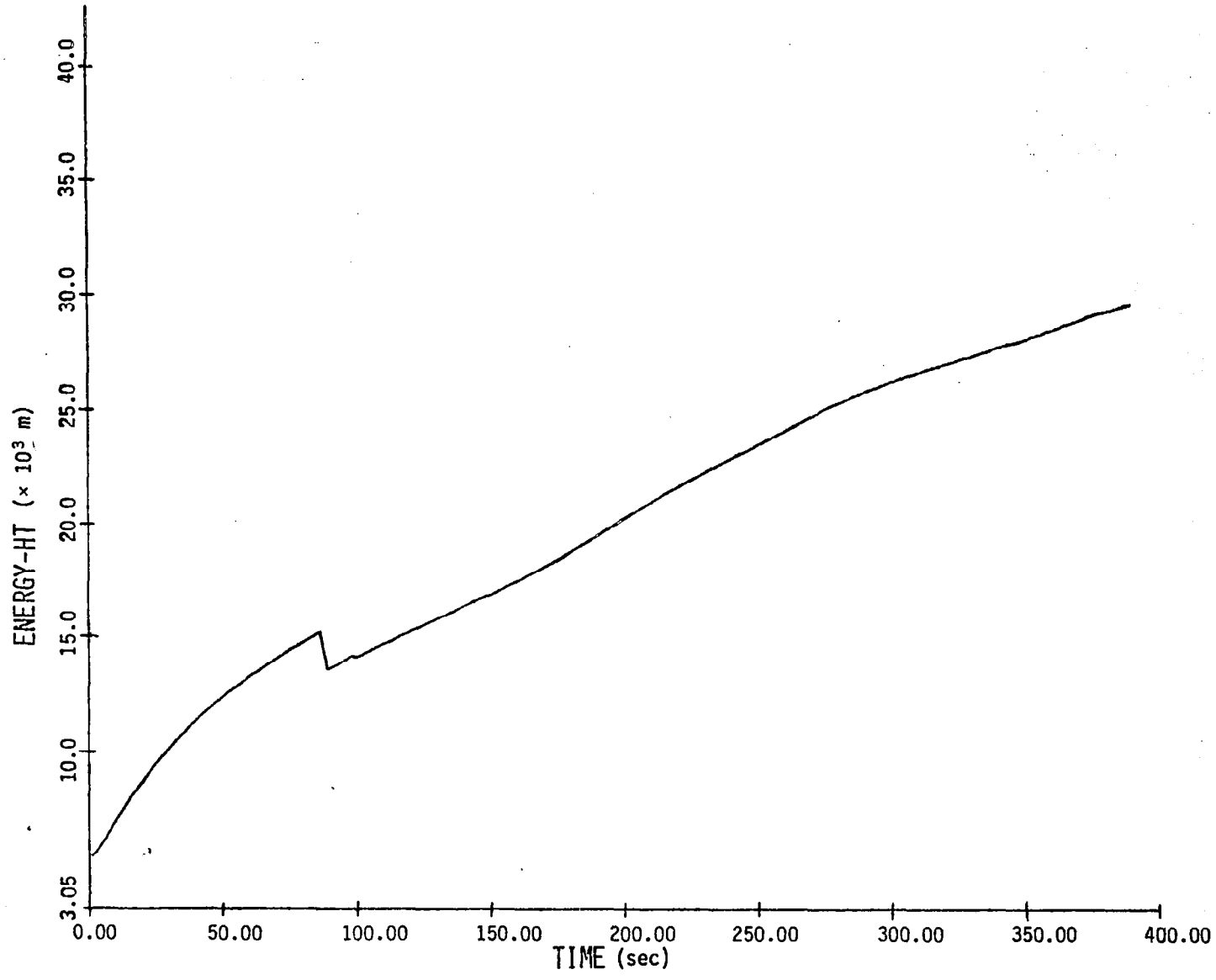


Figure 6.2.20 Energy Time-History

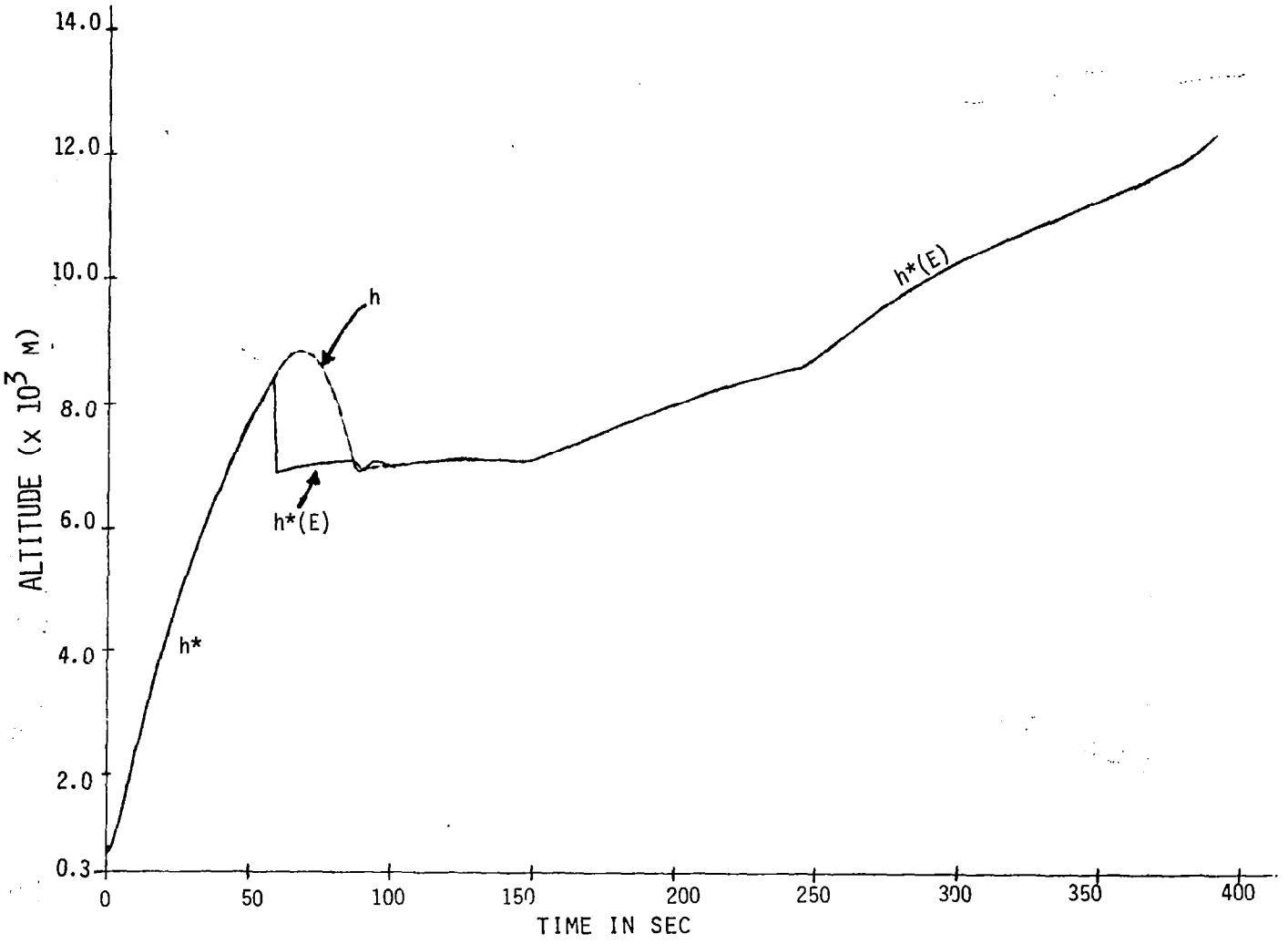


Figure 6.2.21 Altitude Time-History. h*(E) denotes the desired altitude history from the E-boundary layer.

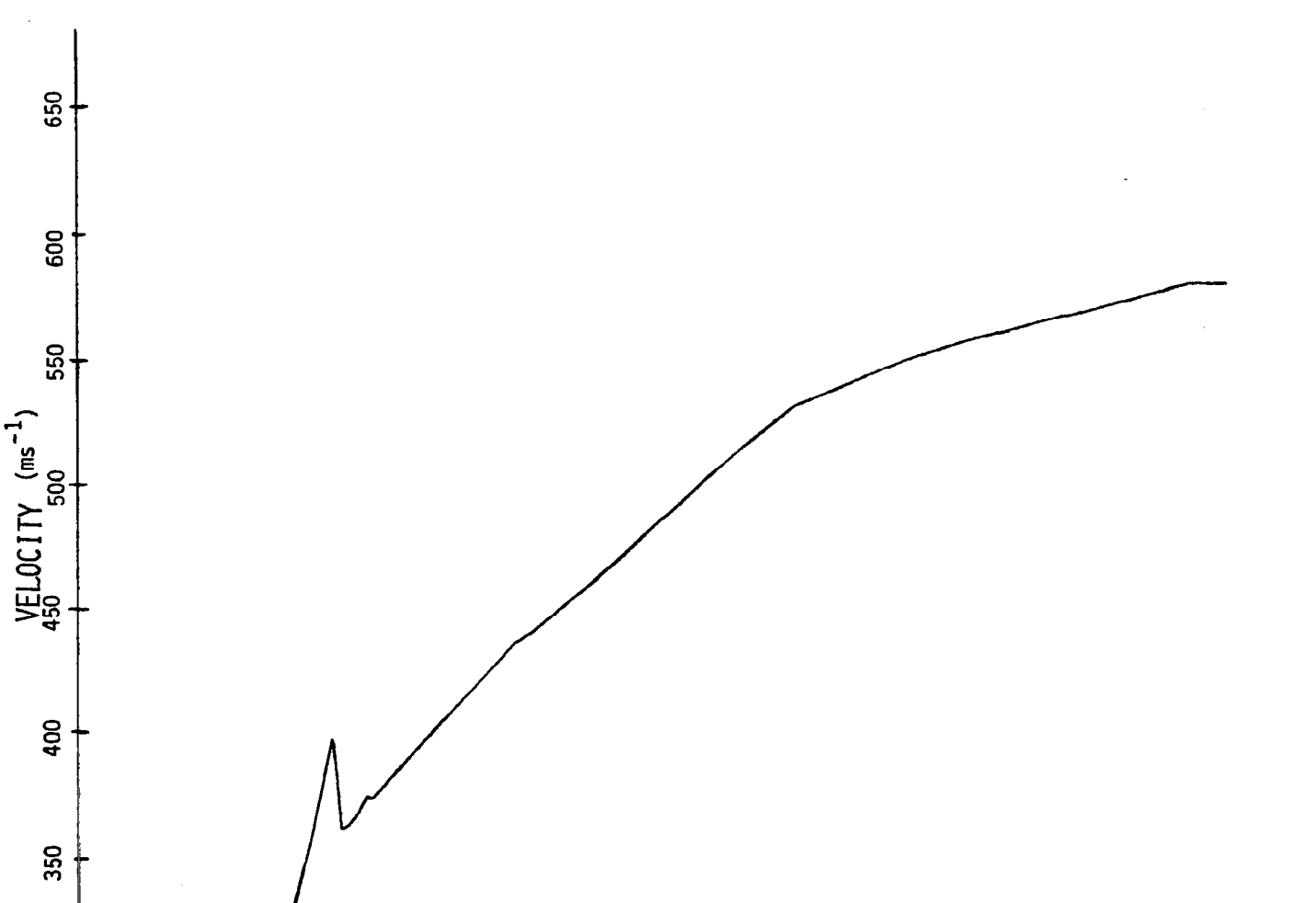


Figure 6.2.22 Changes in Velocity with Time

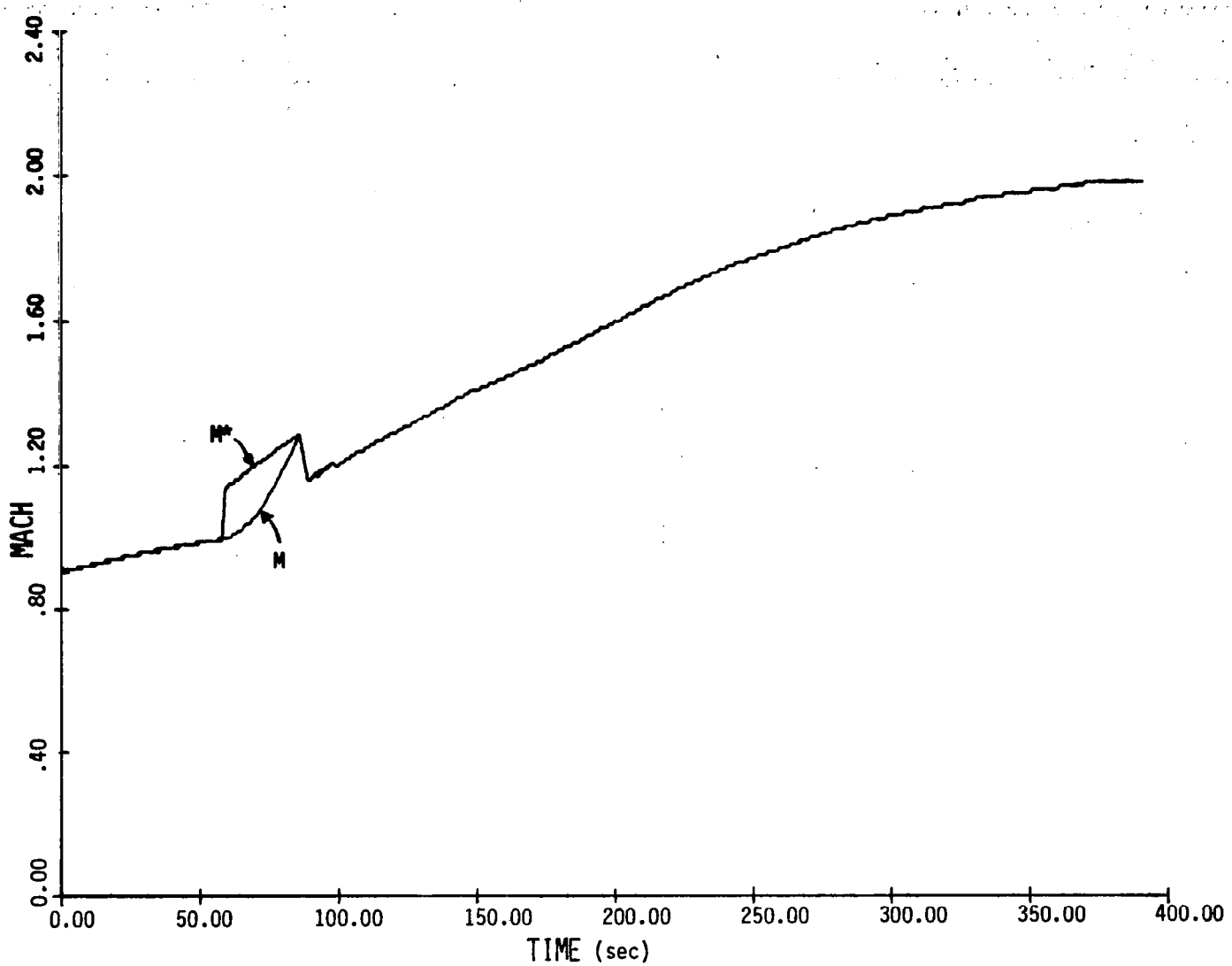


Figure 6.2.23 Variations in Mach Number with Time. M* denotes Mach number computed from the E-boundary layer.

Figure 6.2.24 Changes in Thrust with Time. (Note: $T = T_{max}$ everywhere.)

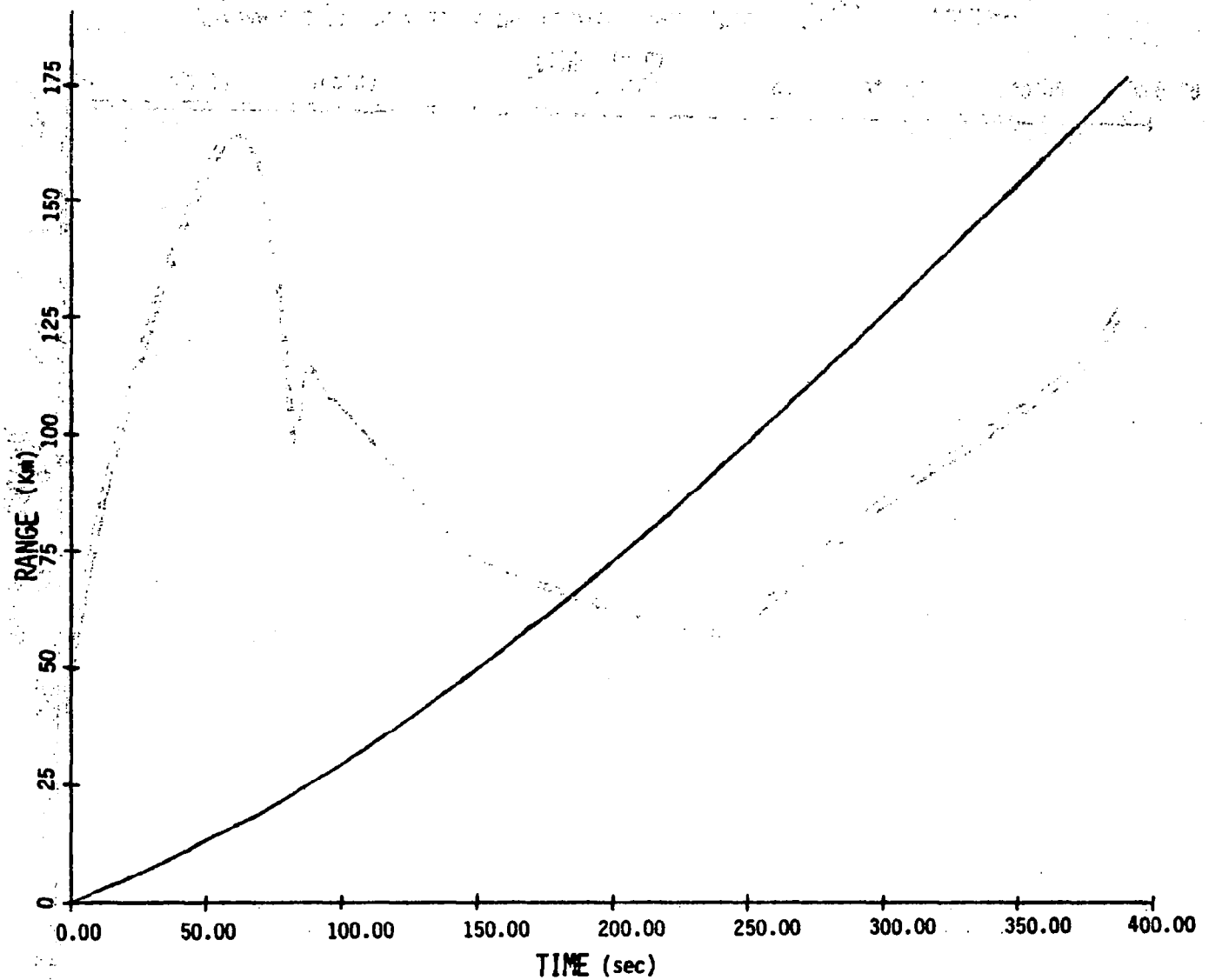


Figure 6.2.25 Range in Horizontal Plane with Time

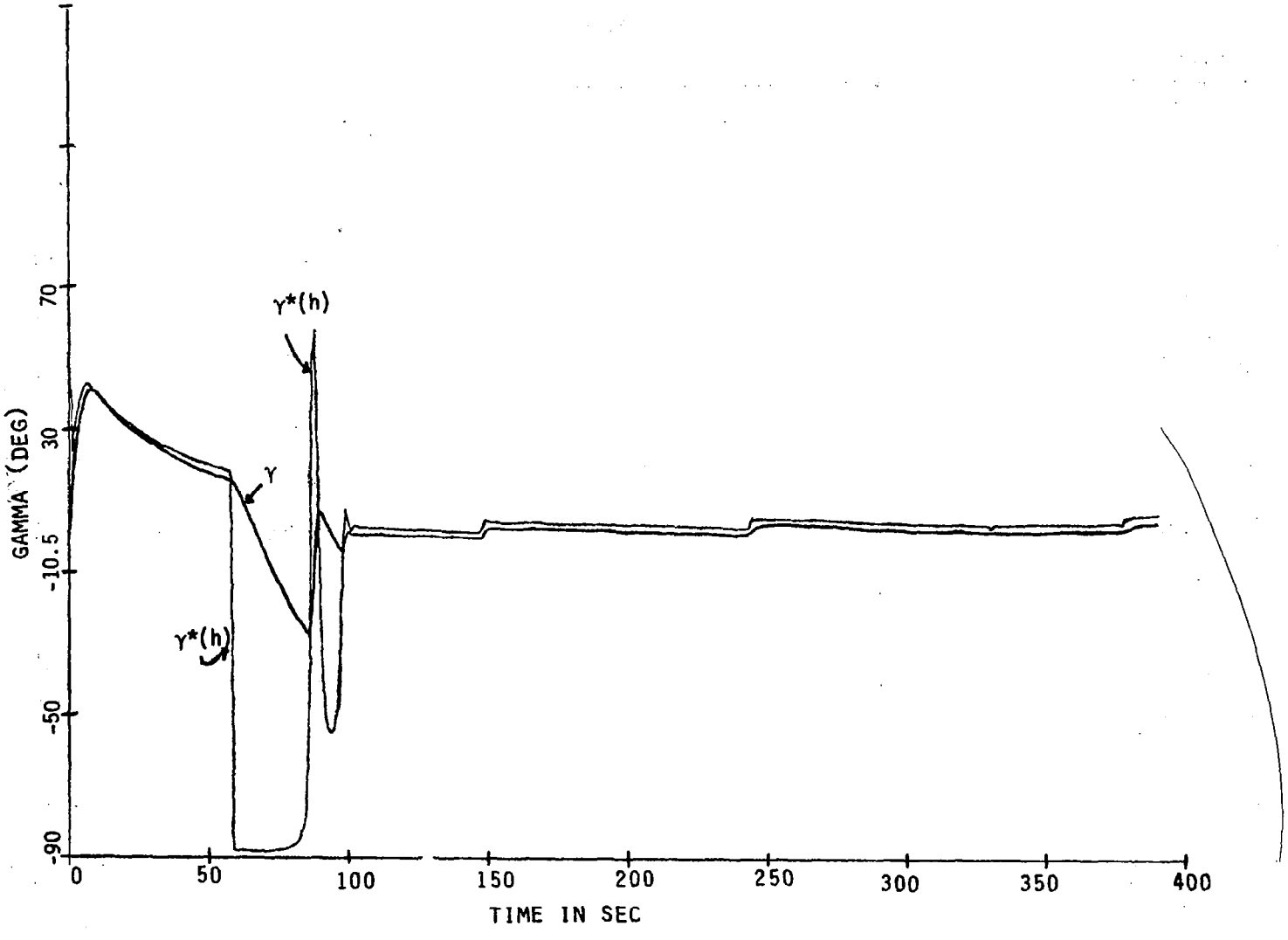


Figure 6.2.26 γ-Time Histories. γ*(h) denotes the desired γ-history computed from the h-boundary layer.

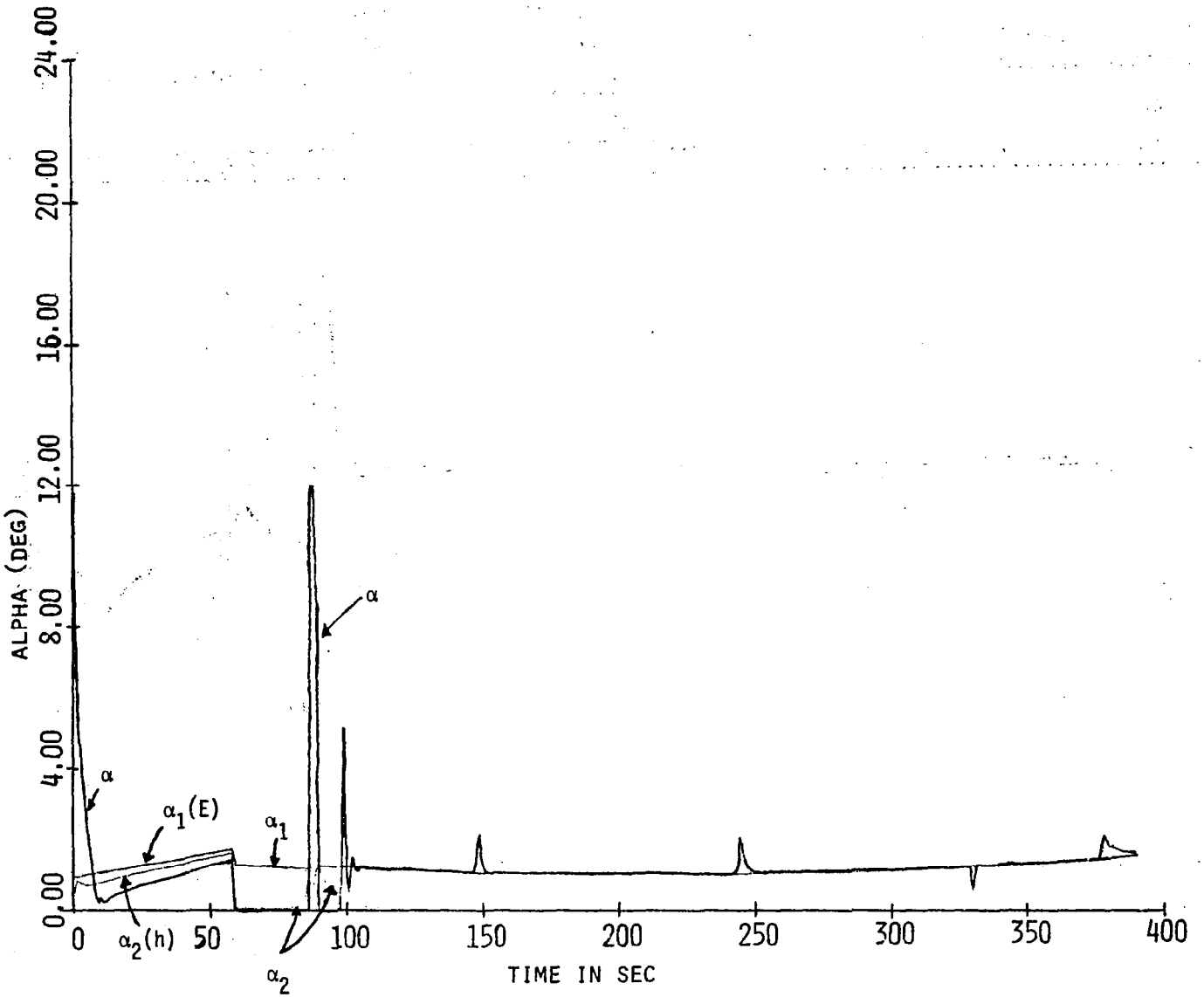


Figure 6.2.27 α-Control Histories. $\alpha_1(E)$ is the control history computed by the energy boundary layer. $\alpha_2(h)$ is the control history computed by the h-boundary layer. The actual control used by the aircraft is α.

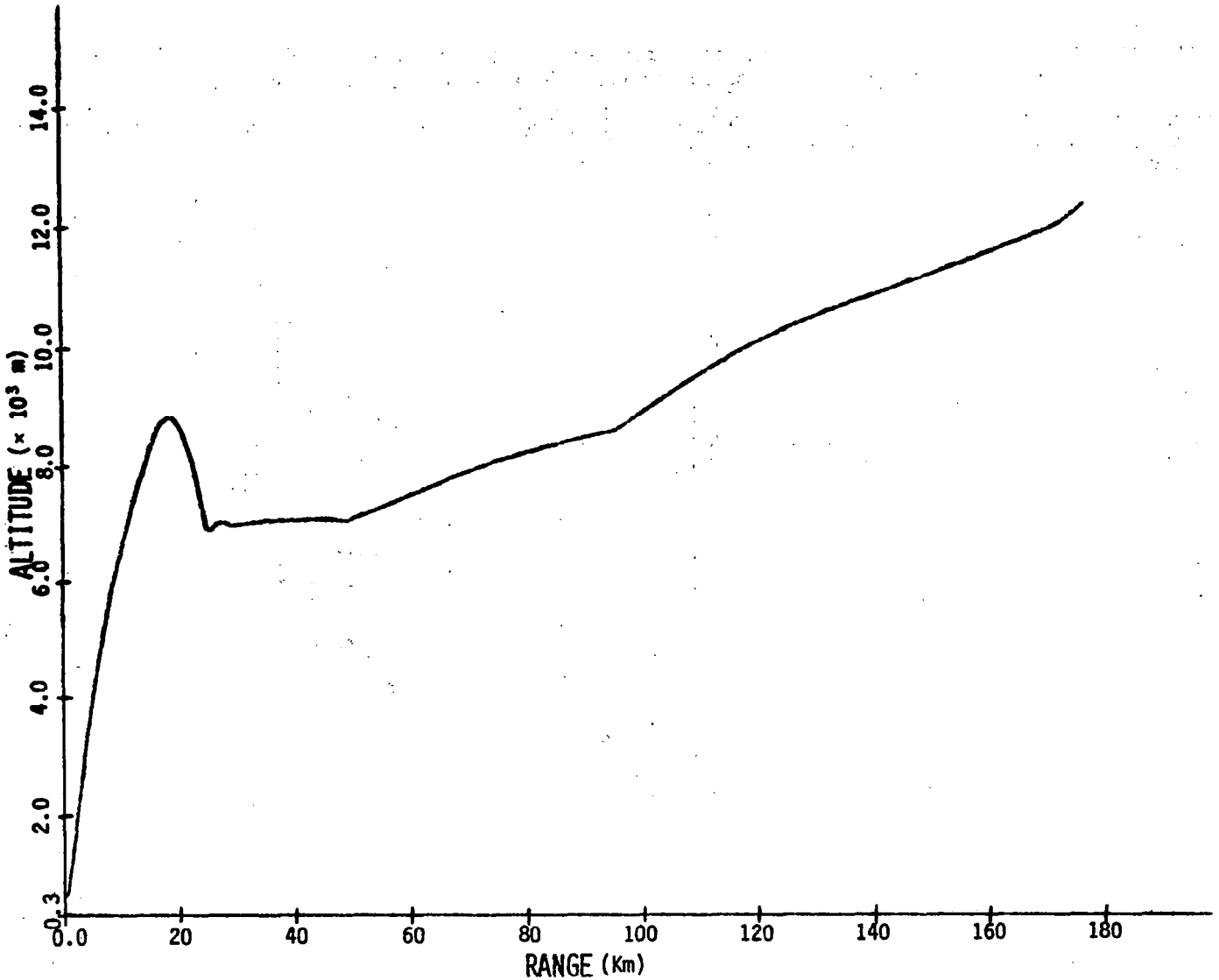


Figure 6.2.28 Altitude-Range Profile of Trajectory

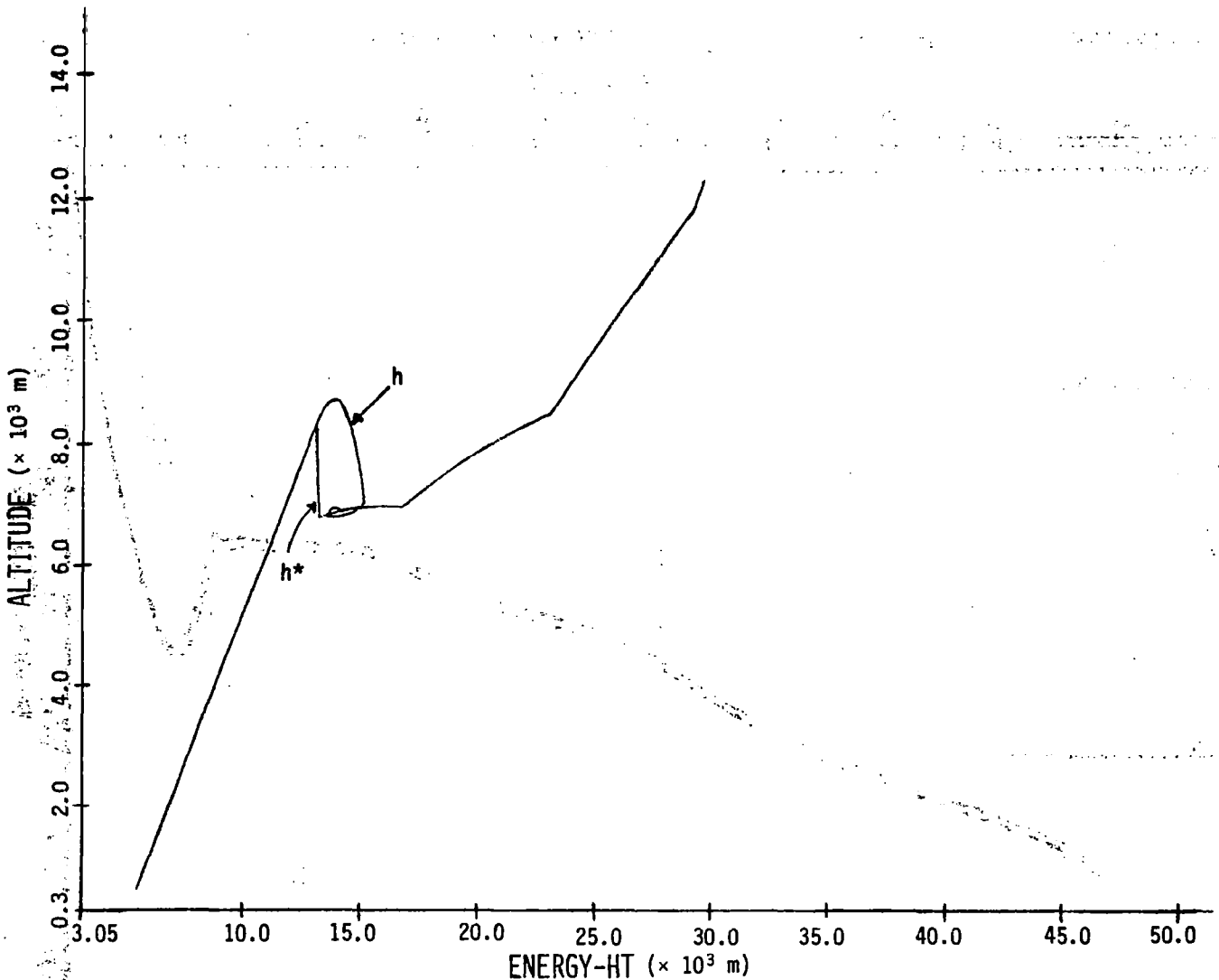


Figure 6.2.29 Altitude-Energy Profile. *h** denotes the altitude profile computed by the E-boundary layer.

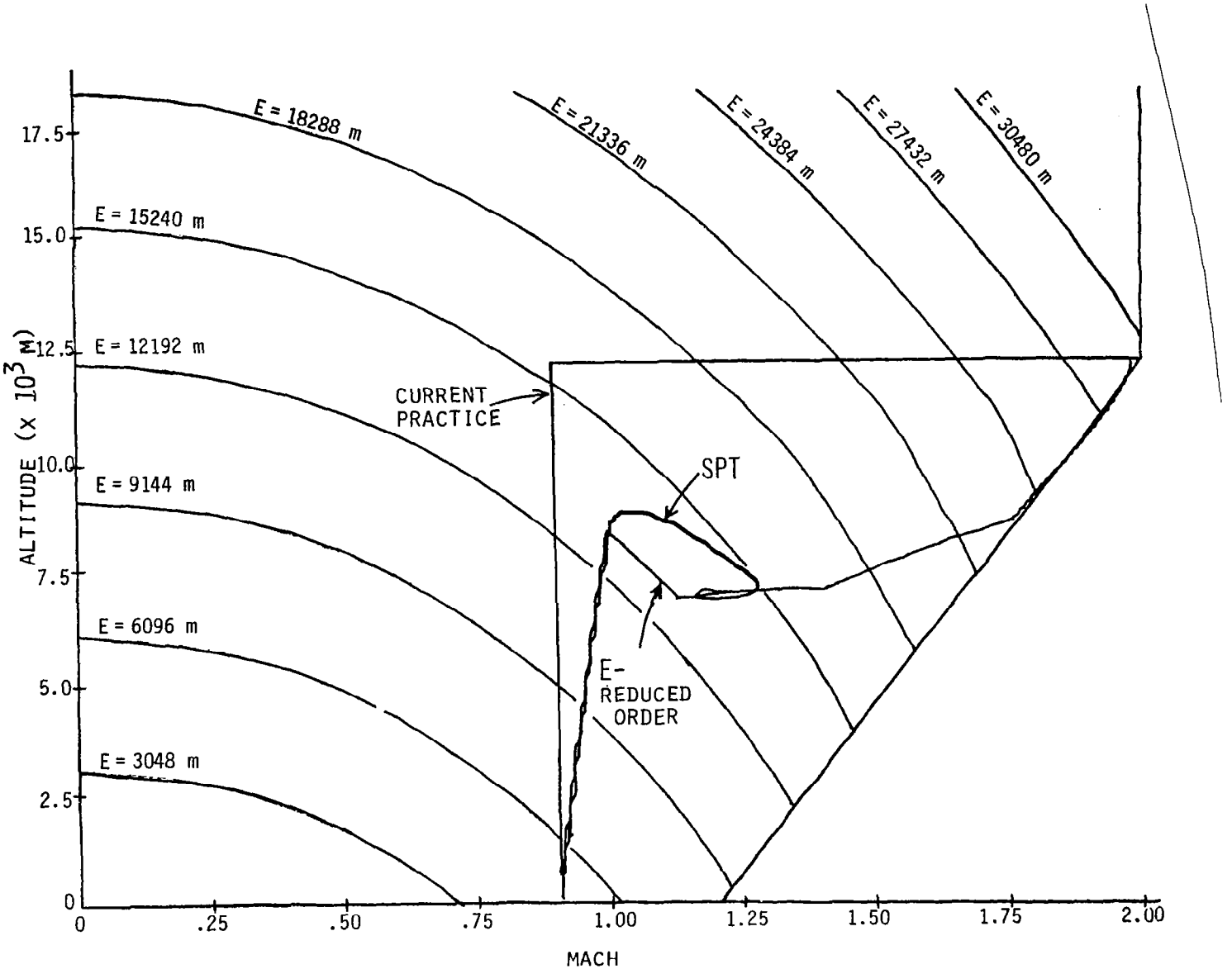


Figure 6.2.30 Altitude-Mach Profile. Also shown is the E-reduced order solution and the current practice trajectory.

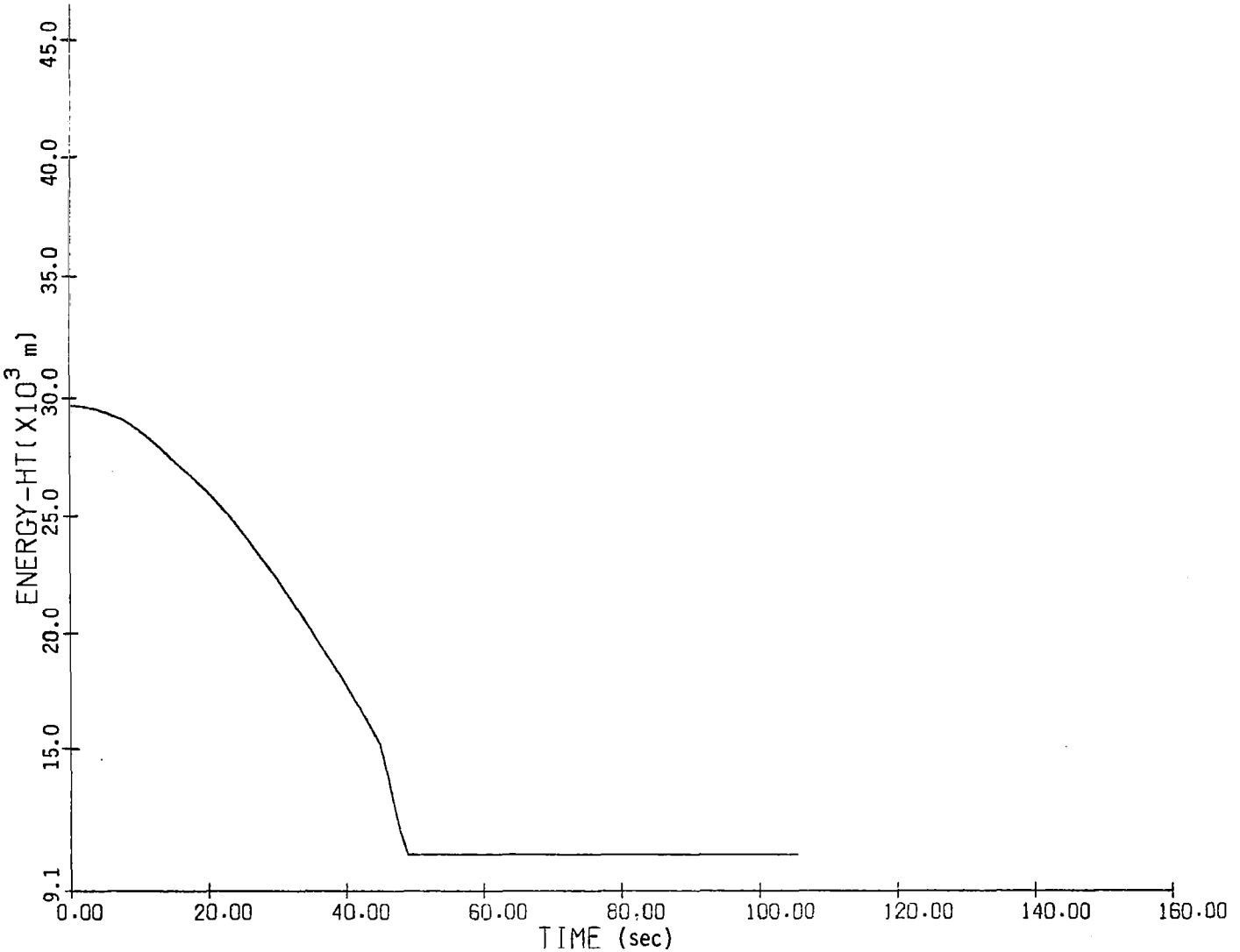


Figure 6.2.31 Energy Time-History

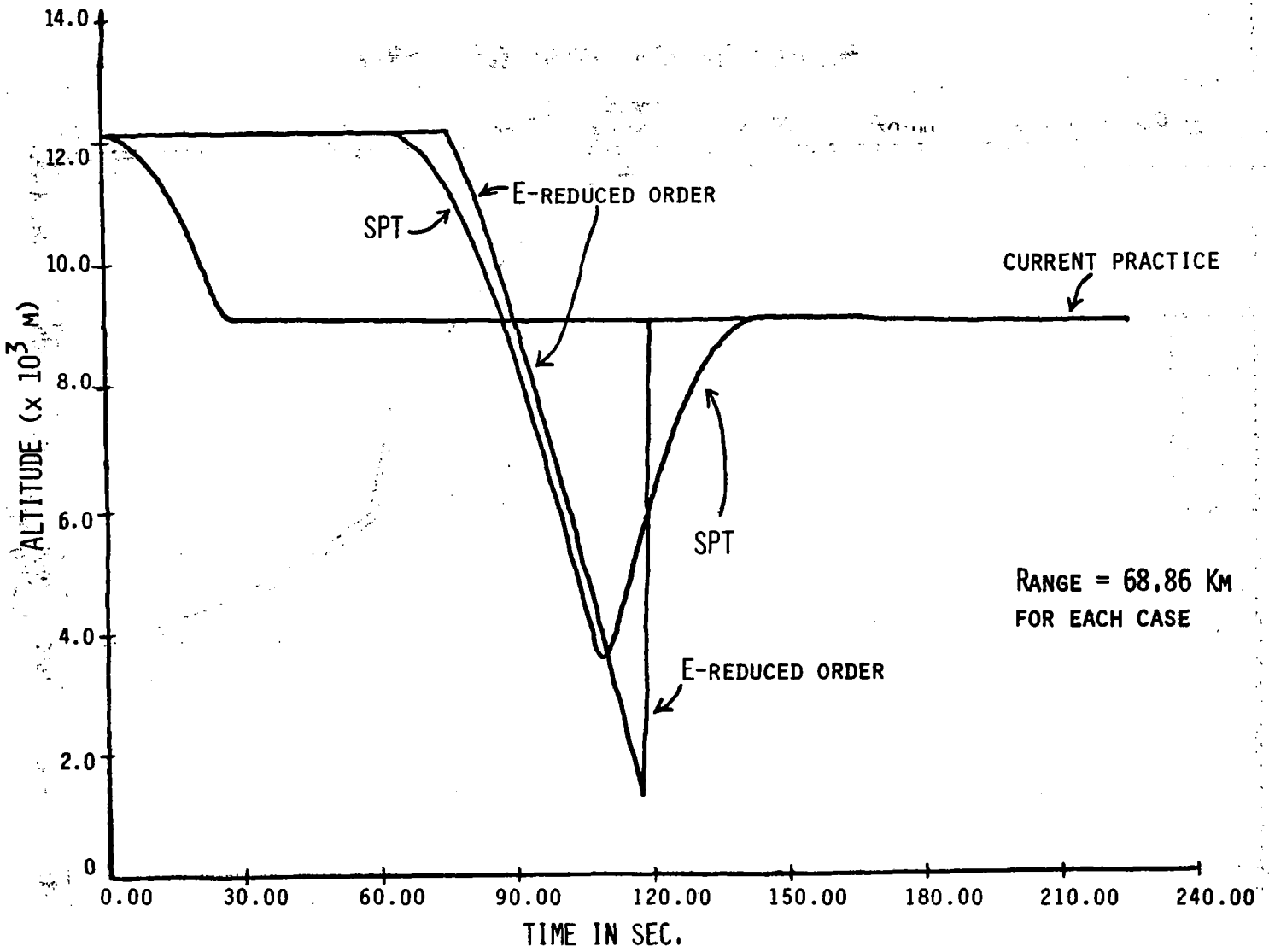


Figure 6.2.32 Variations in Altitude with Time

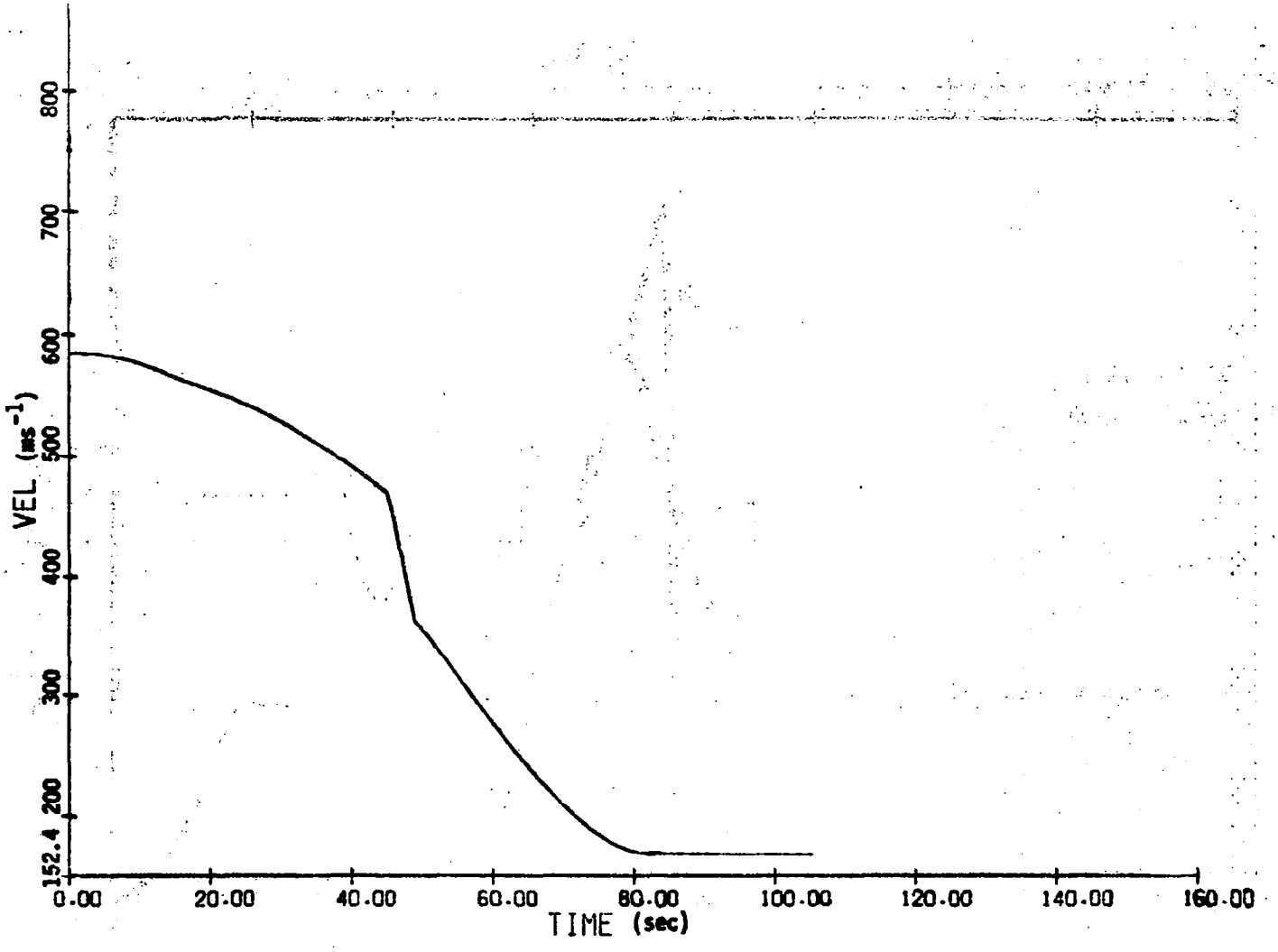


Figure 6.2.33 Changes in Velocity with Time

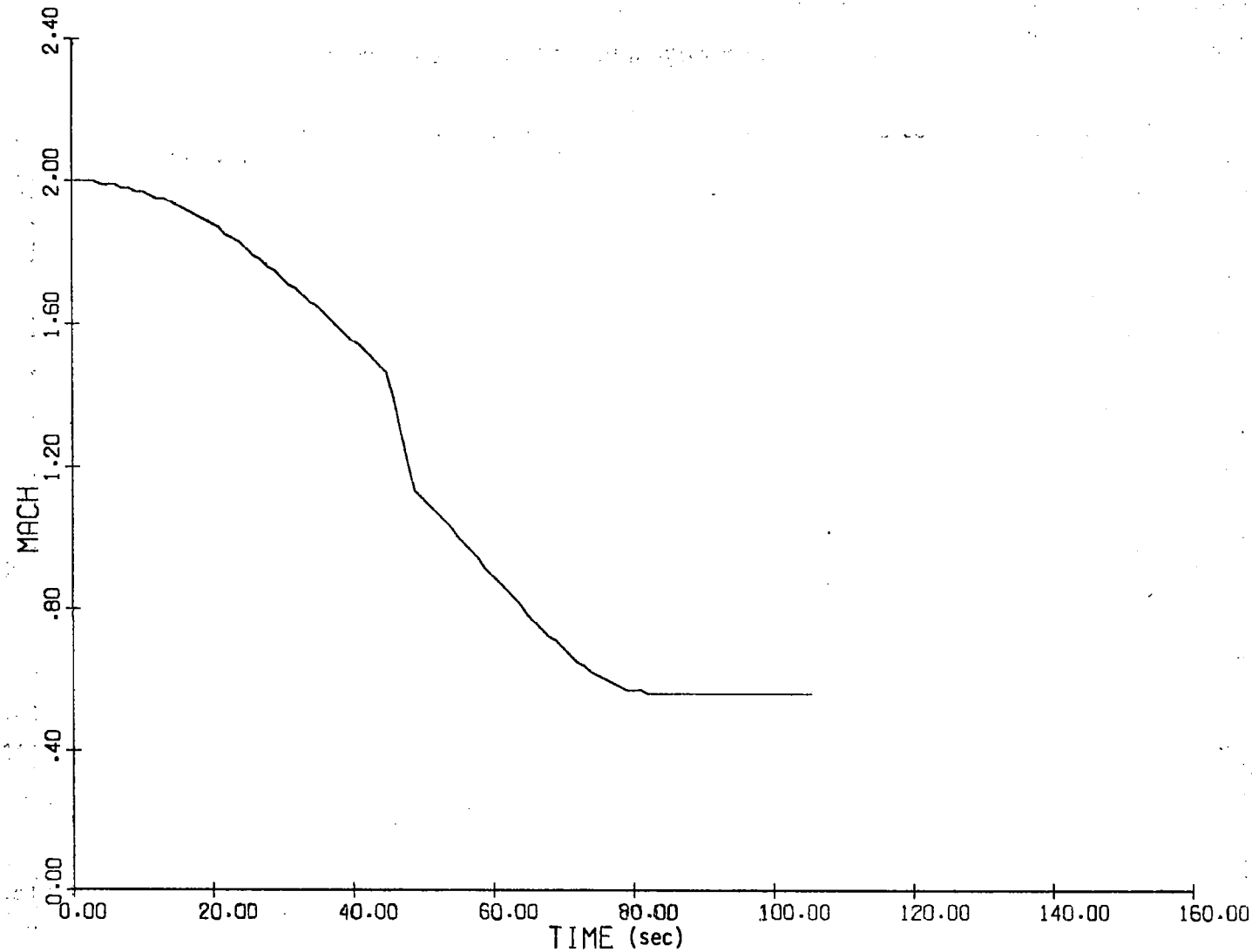


Figure 6.2.34 Variations in Mach Number with Time

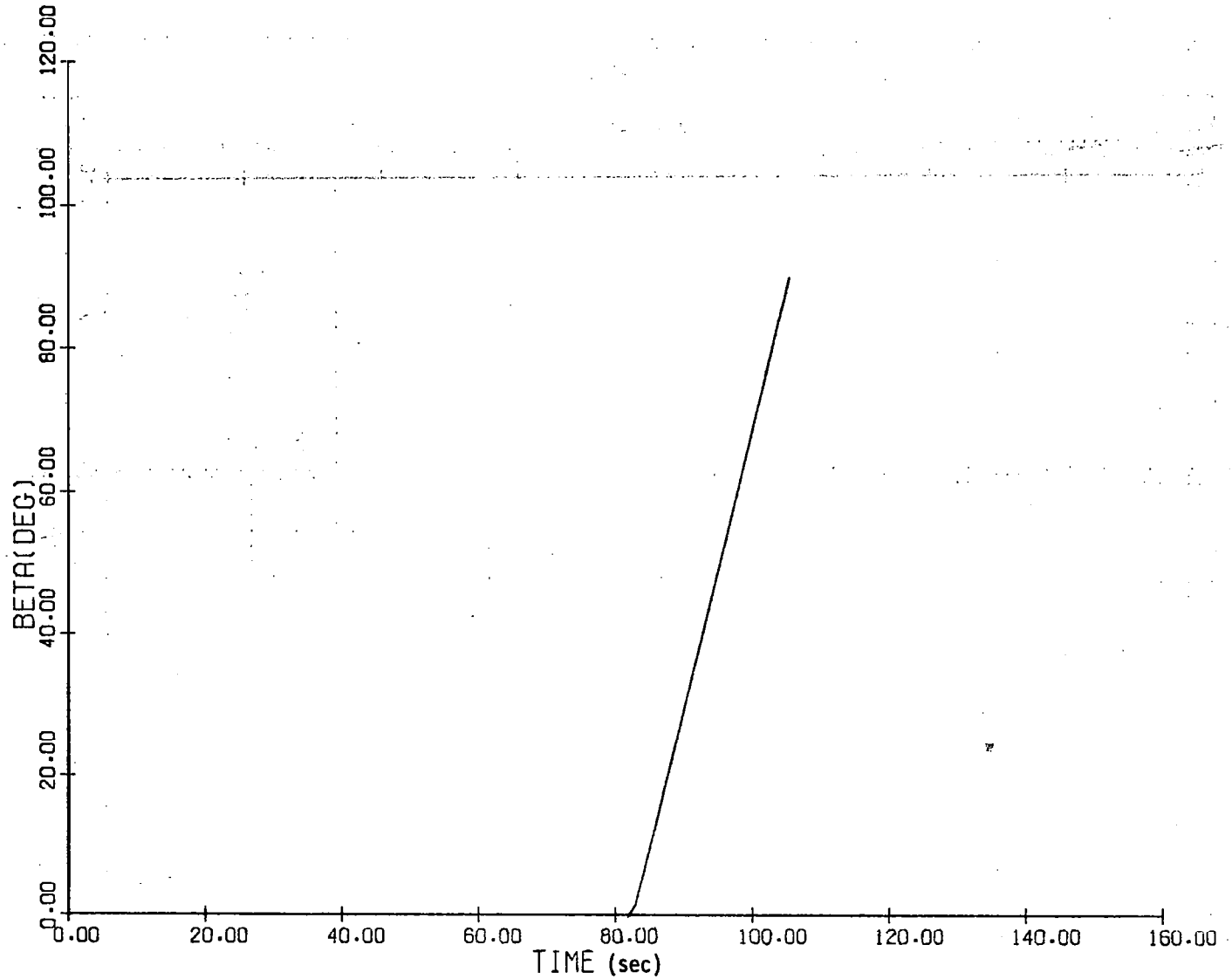


Figure 6.2.35 Changes in Heading Angle with Time

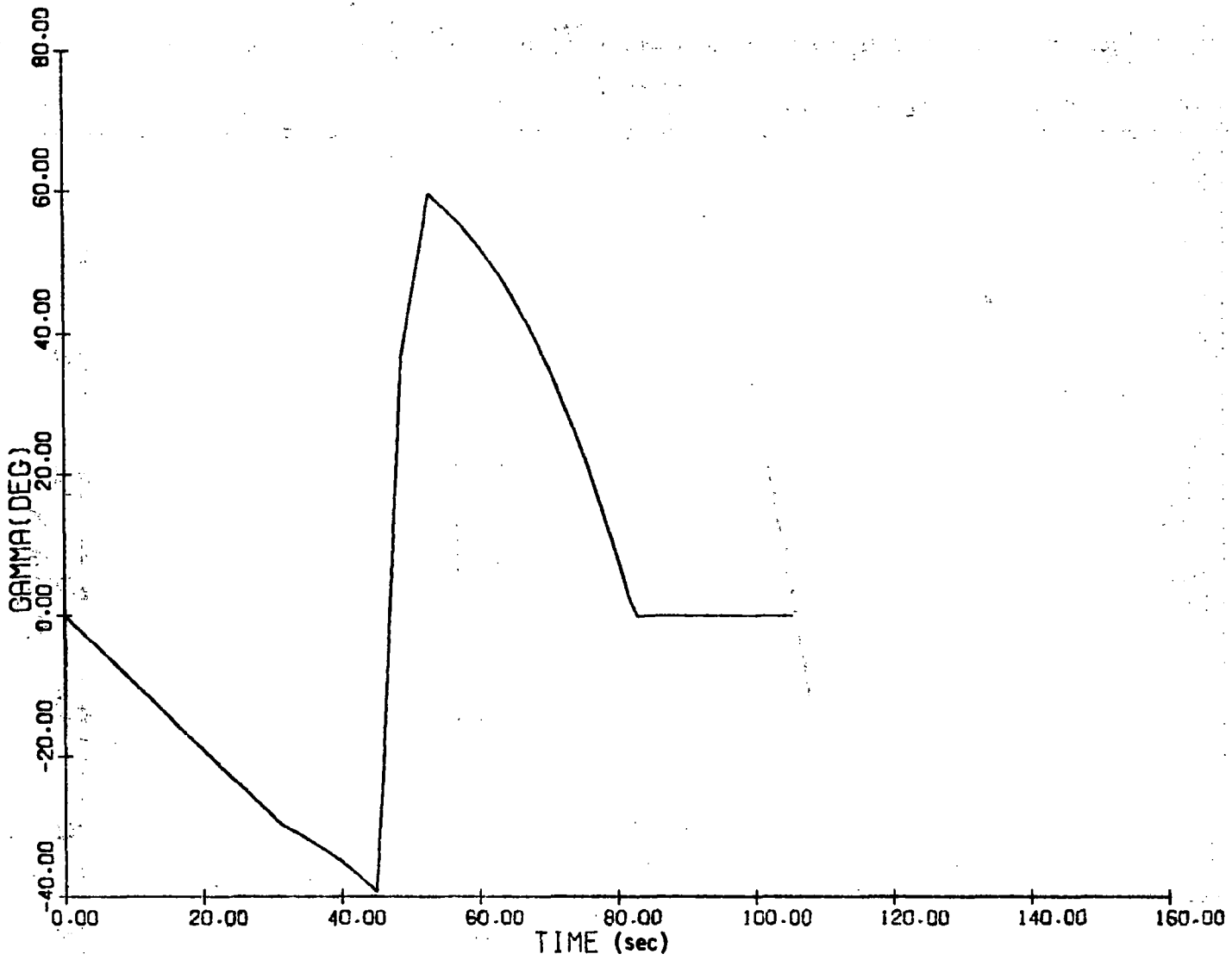


Figure 6.2.36 Gamma Time-History

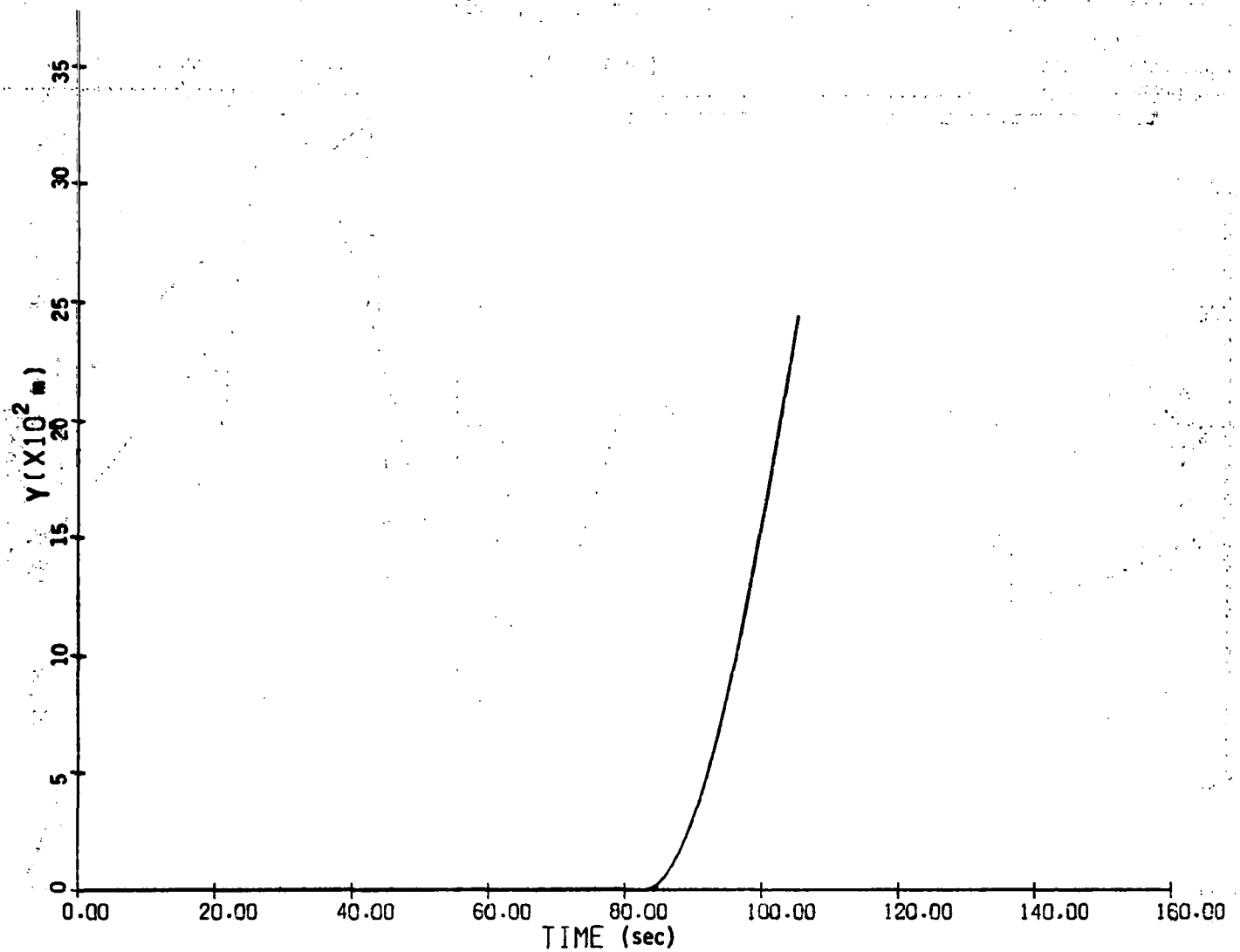


Figure 6.2.37 Variations in the y-component of the Horizontal Plane Projection of the Trajectory

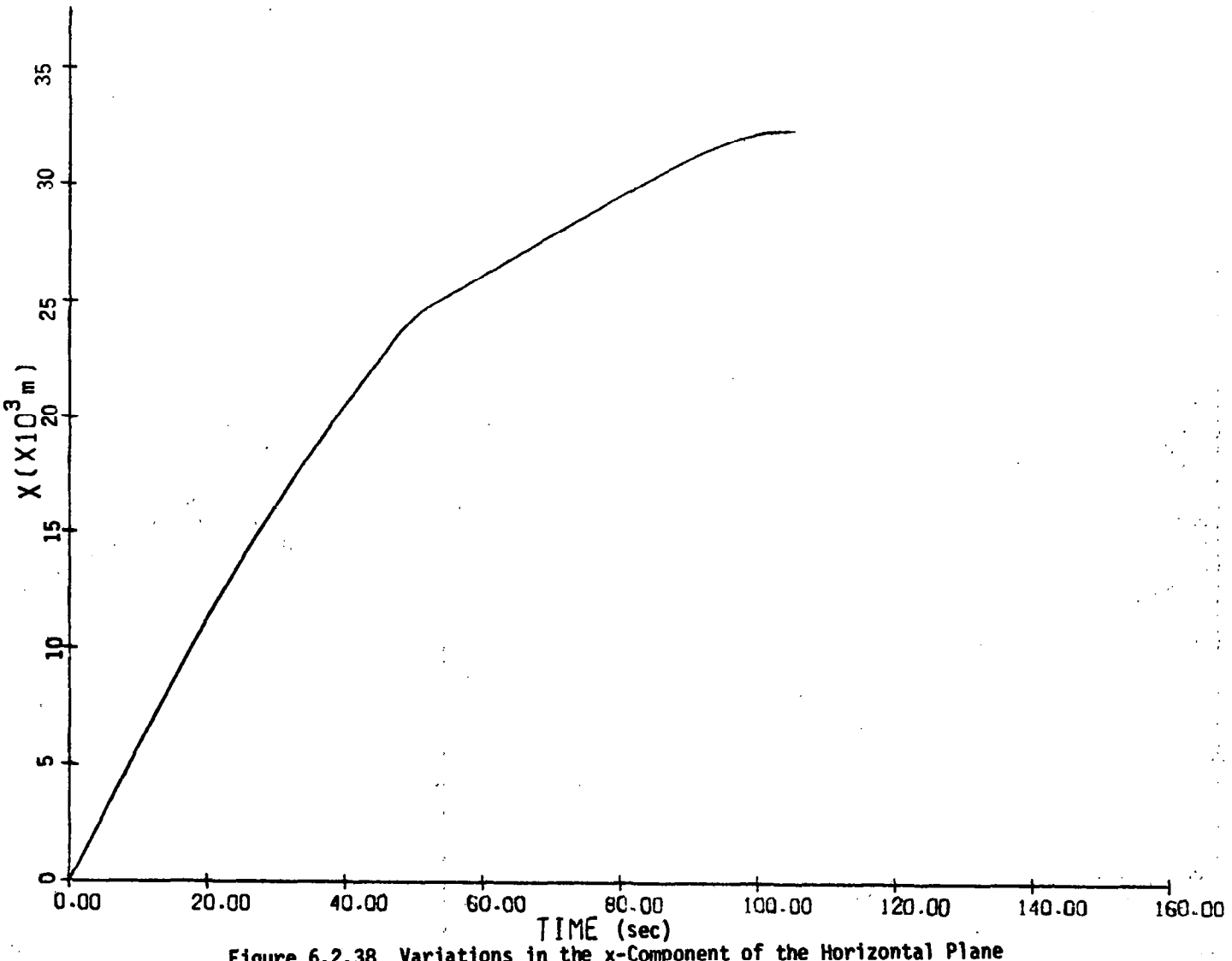


Figure 6.2.38 Variations in the x-Component of the Horizontal Plane Projection of the Trajectory

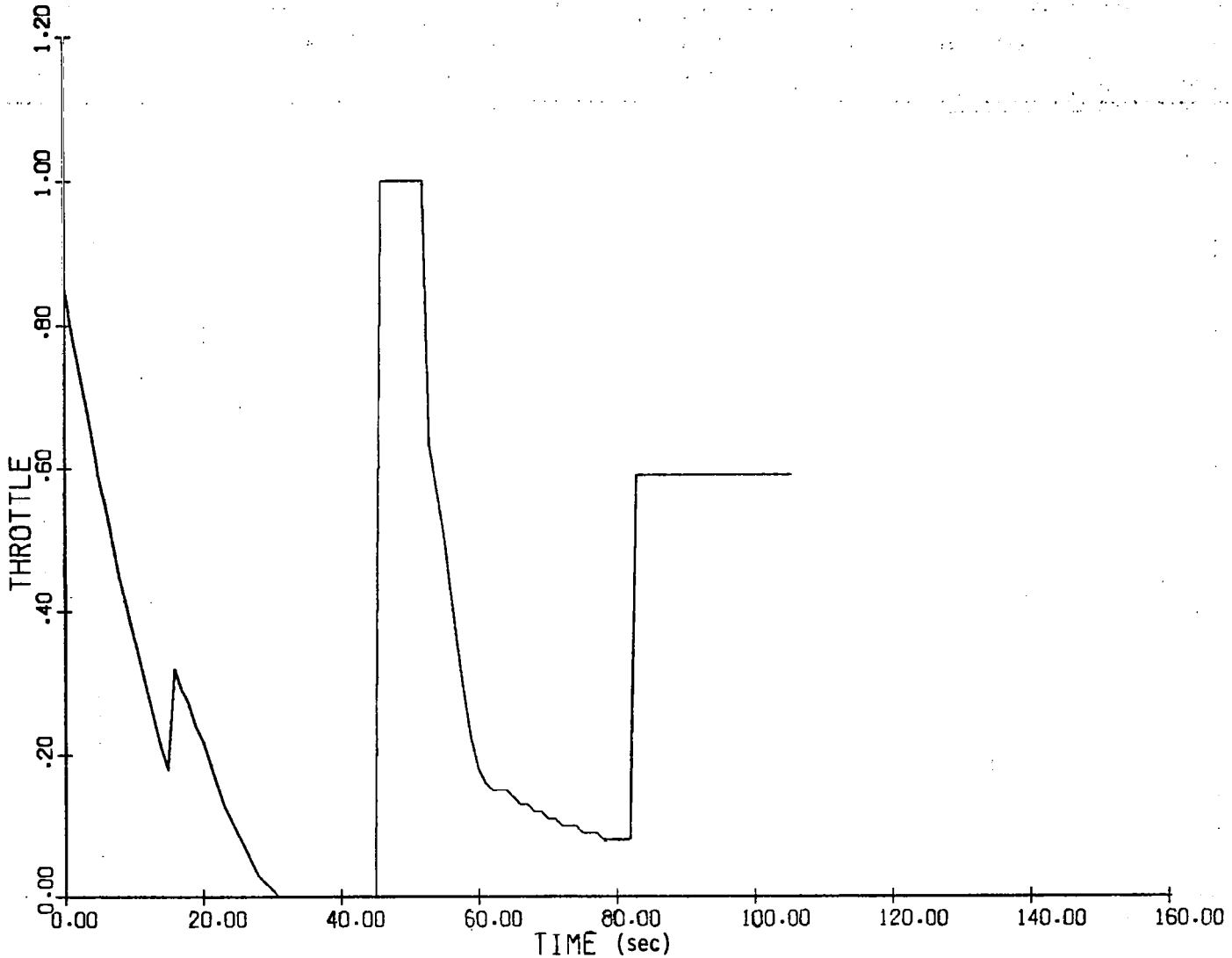


Figure 6.2.39 Throttle Variations with Time

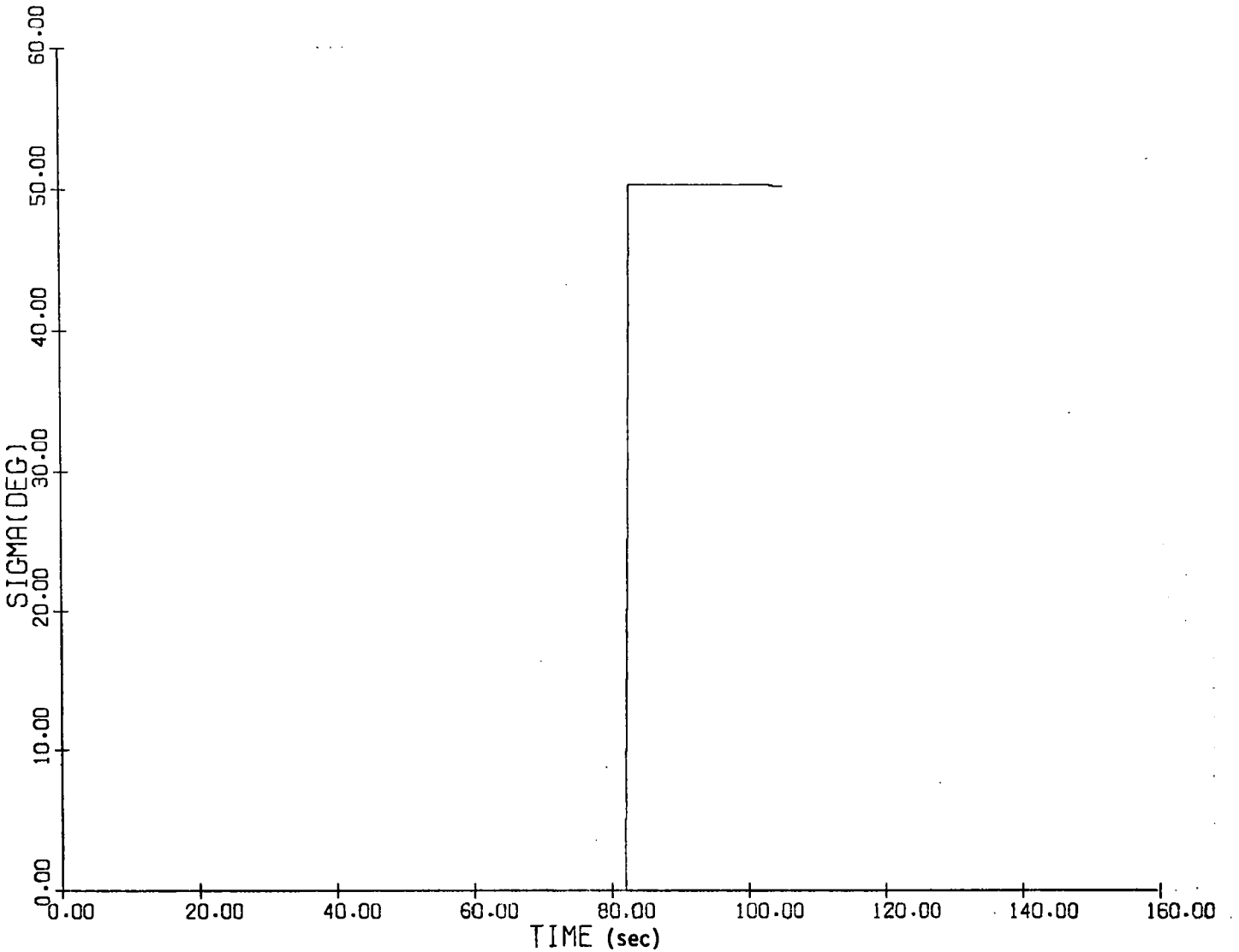


Figure 6.2.40 Bank Angle Control Time-History

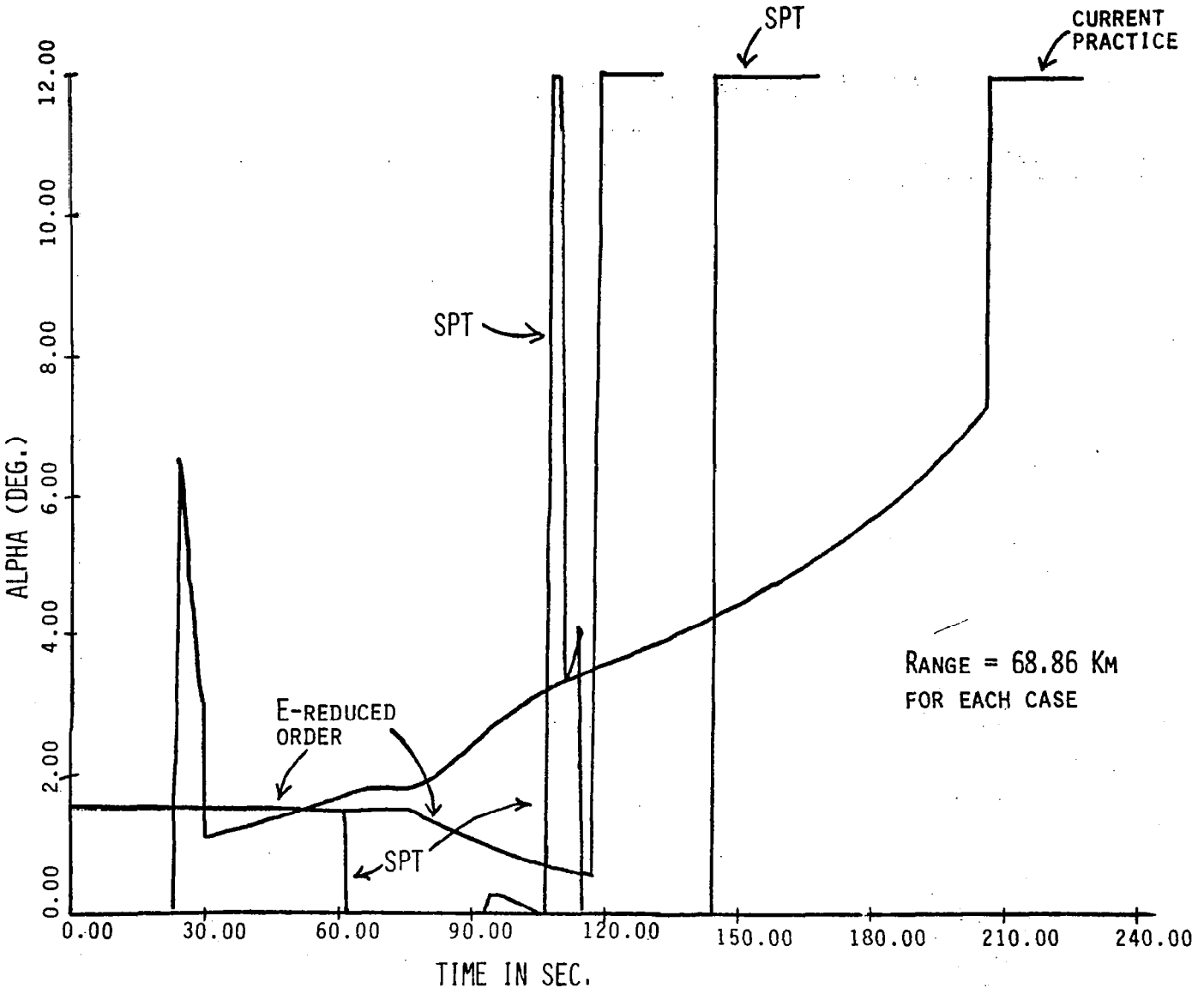


Figure 6.2.41 Alpha Control Histories

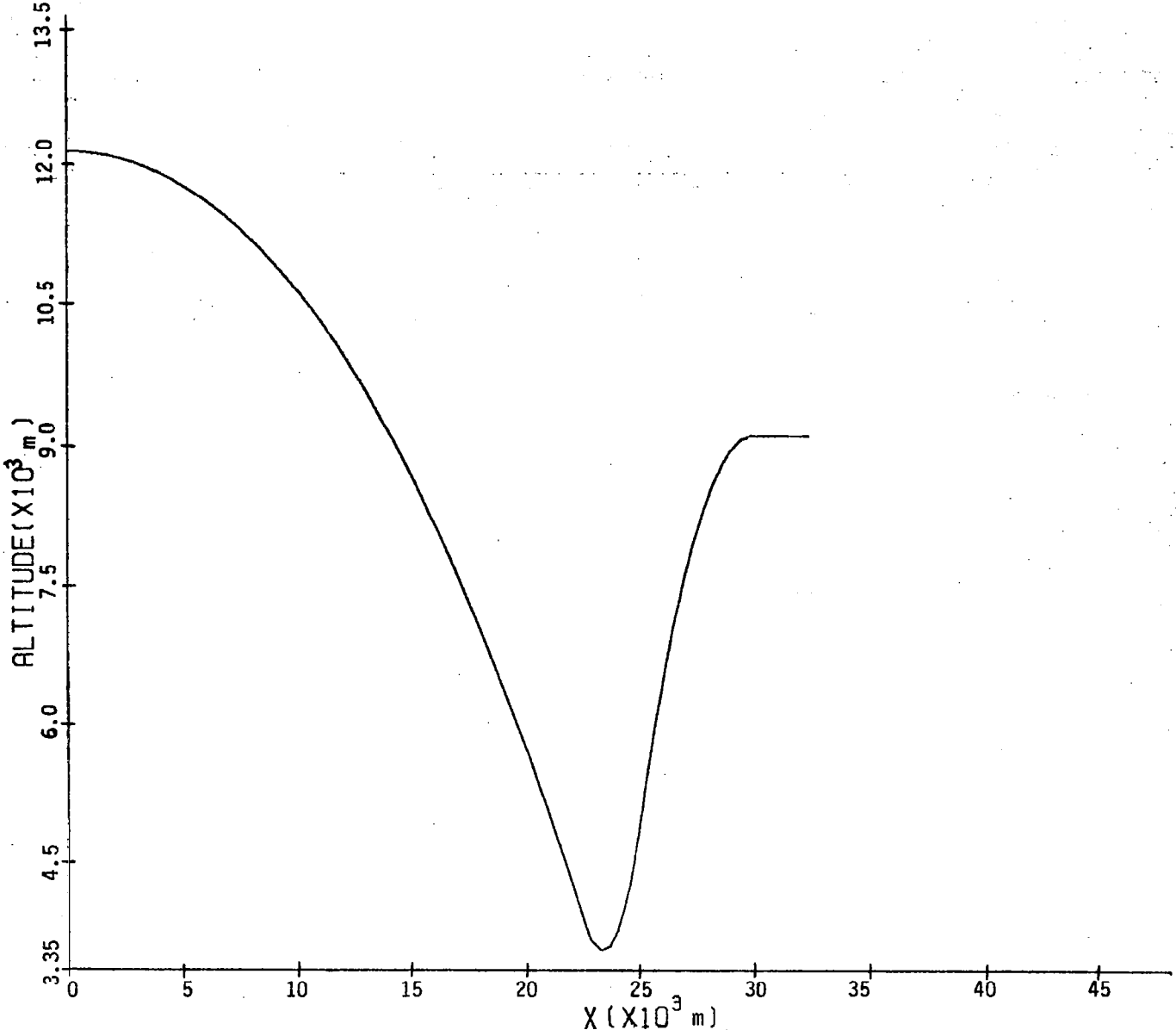


Figure 6.2.42 Vertical Plane Projection of Trajectory. (Note: this is very nearly the altitude-range profile since only a short distance is moved in the y direction while turning at the end).

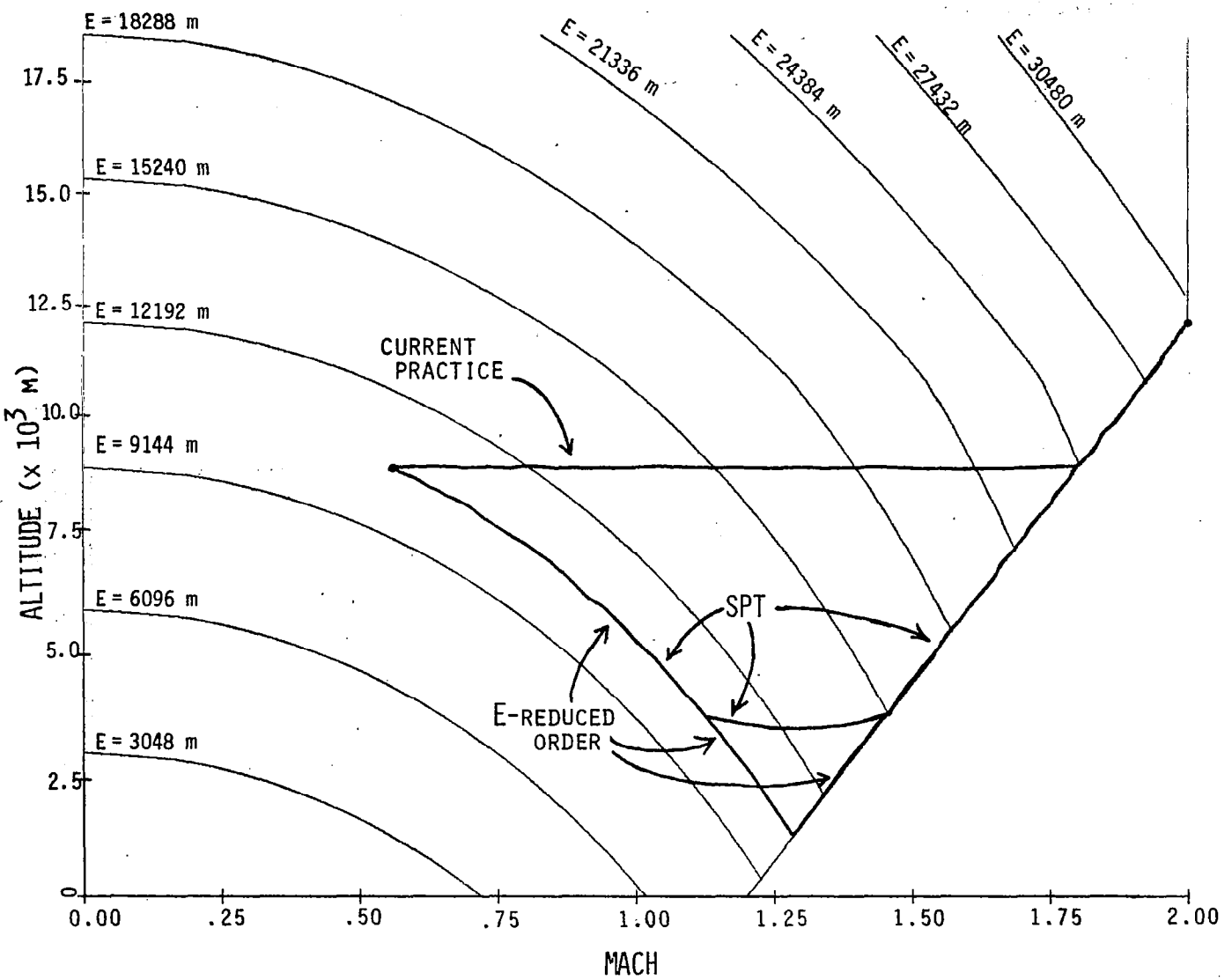


Figure 6.2.43 Altitude - Mach Profile of Trajectory

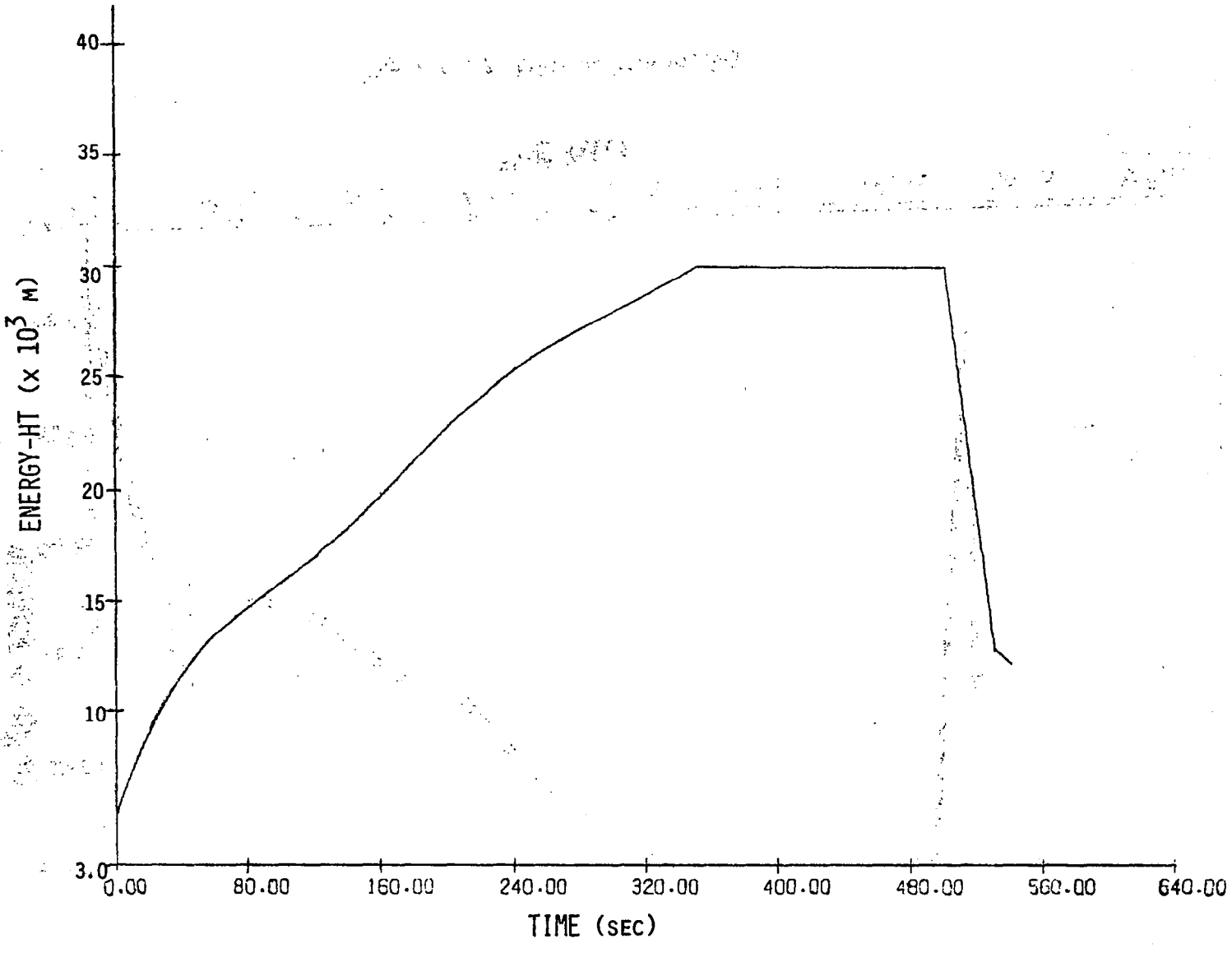


Figure 6.3.1 Changes in Energy with Time

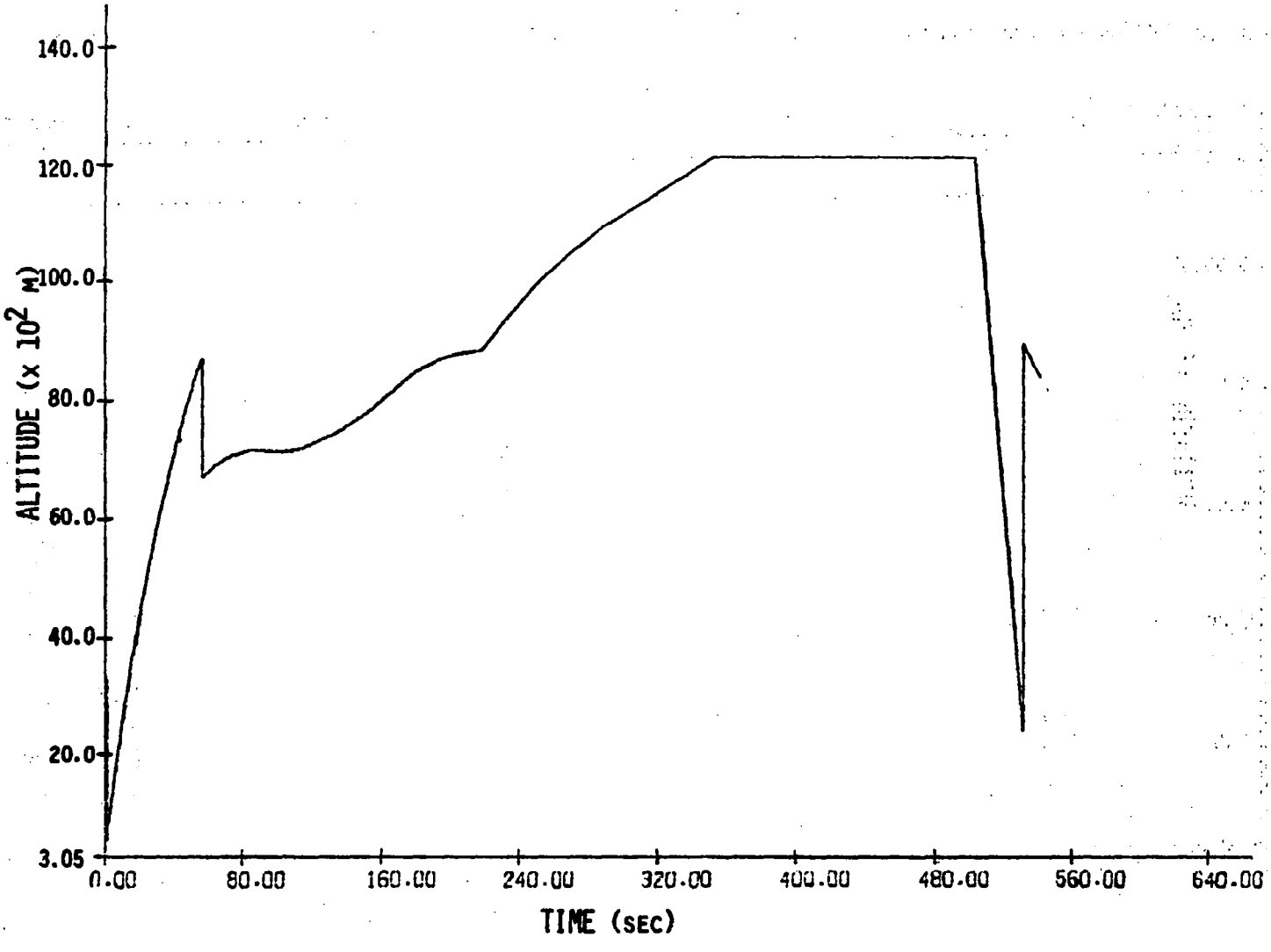


Figure 6.3.2 Altitude Time-History

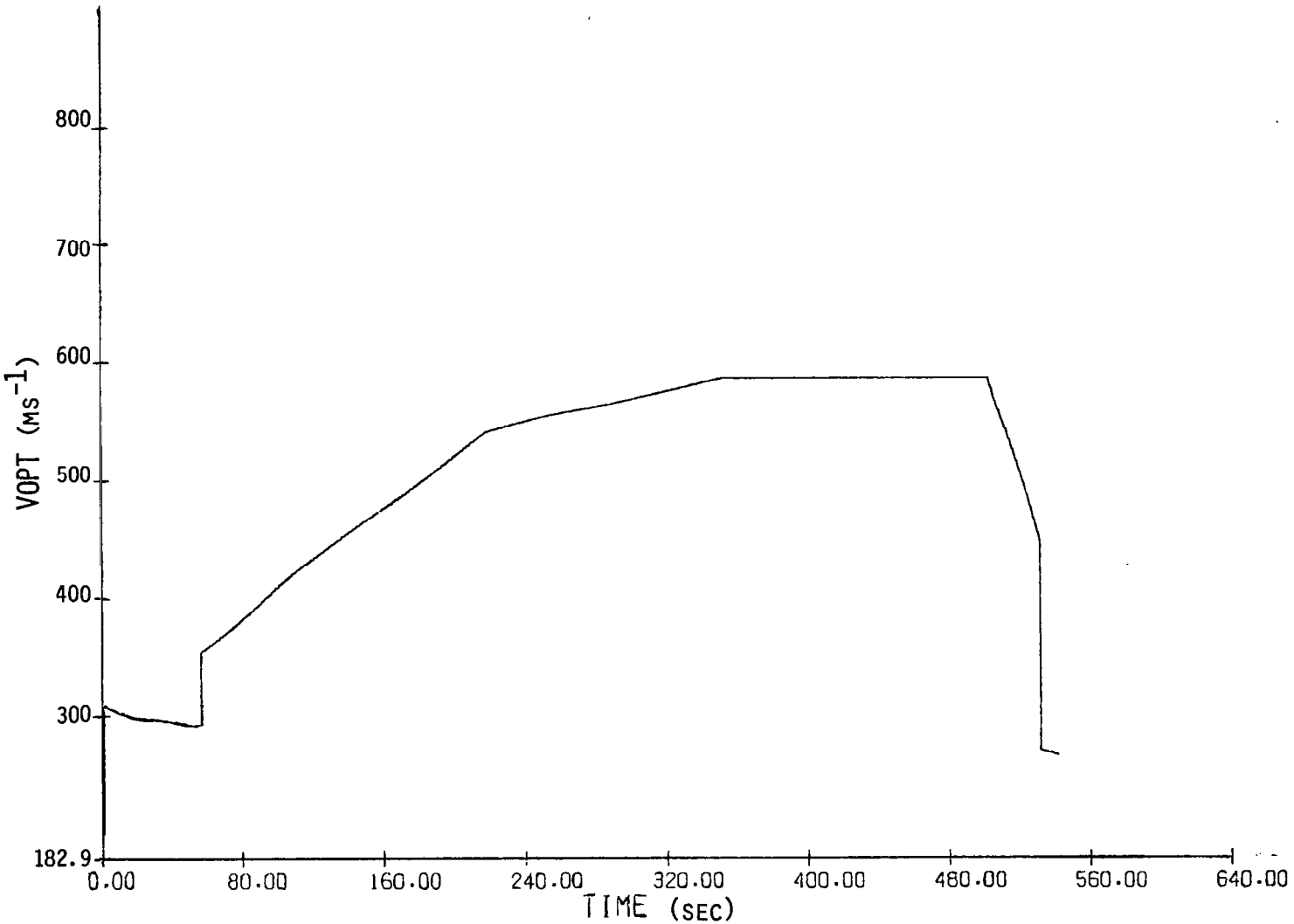


Figure 6.3.3 Changes in Velocity with Time

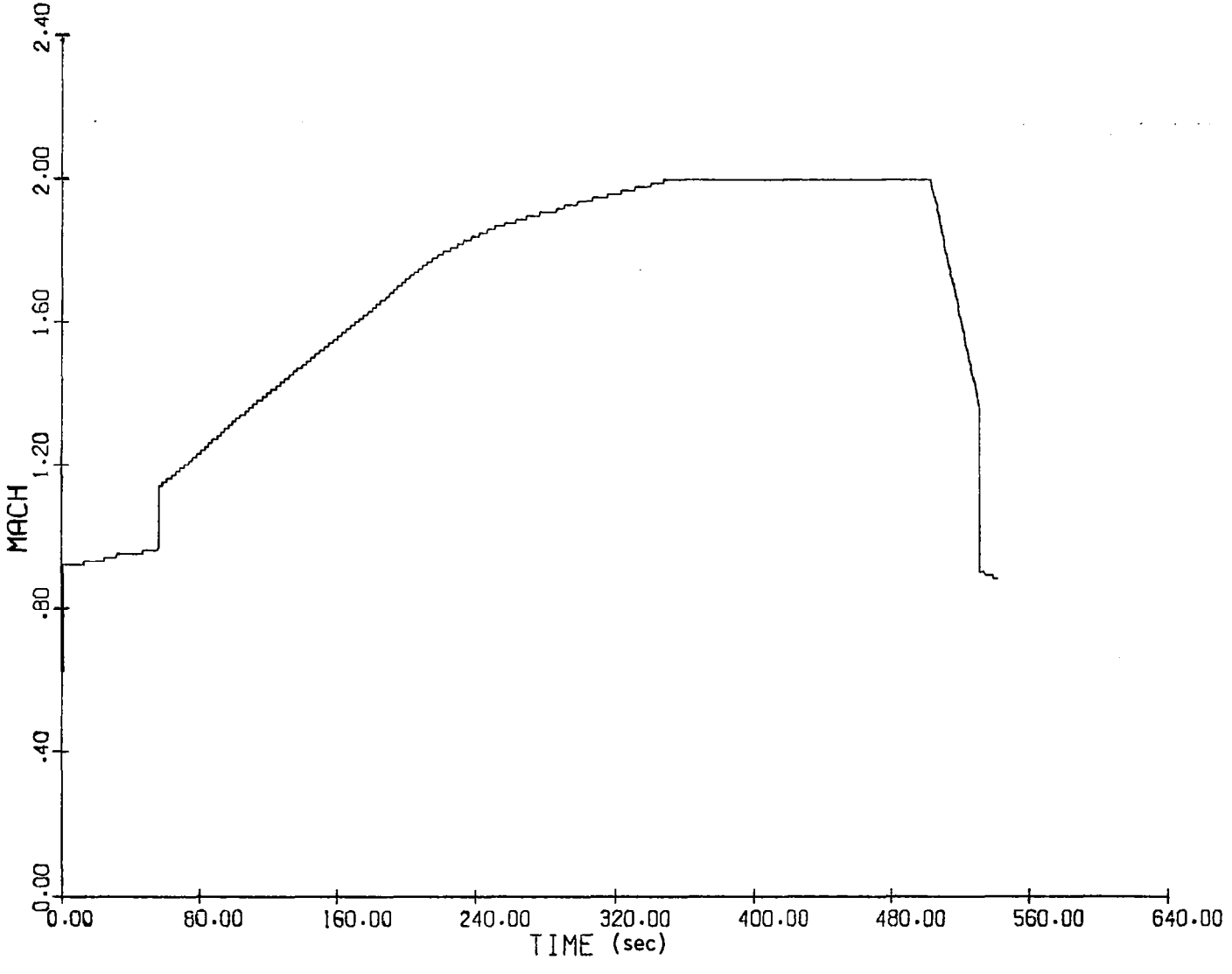


Figure 6.3.4 Changes in Mach Number with Time

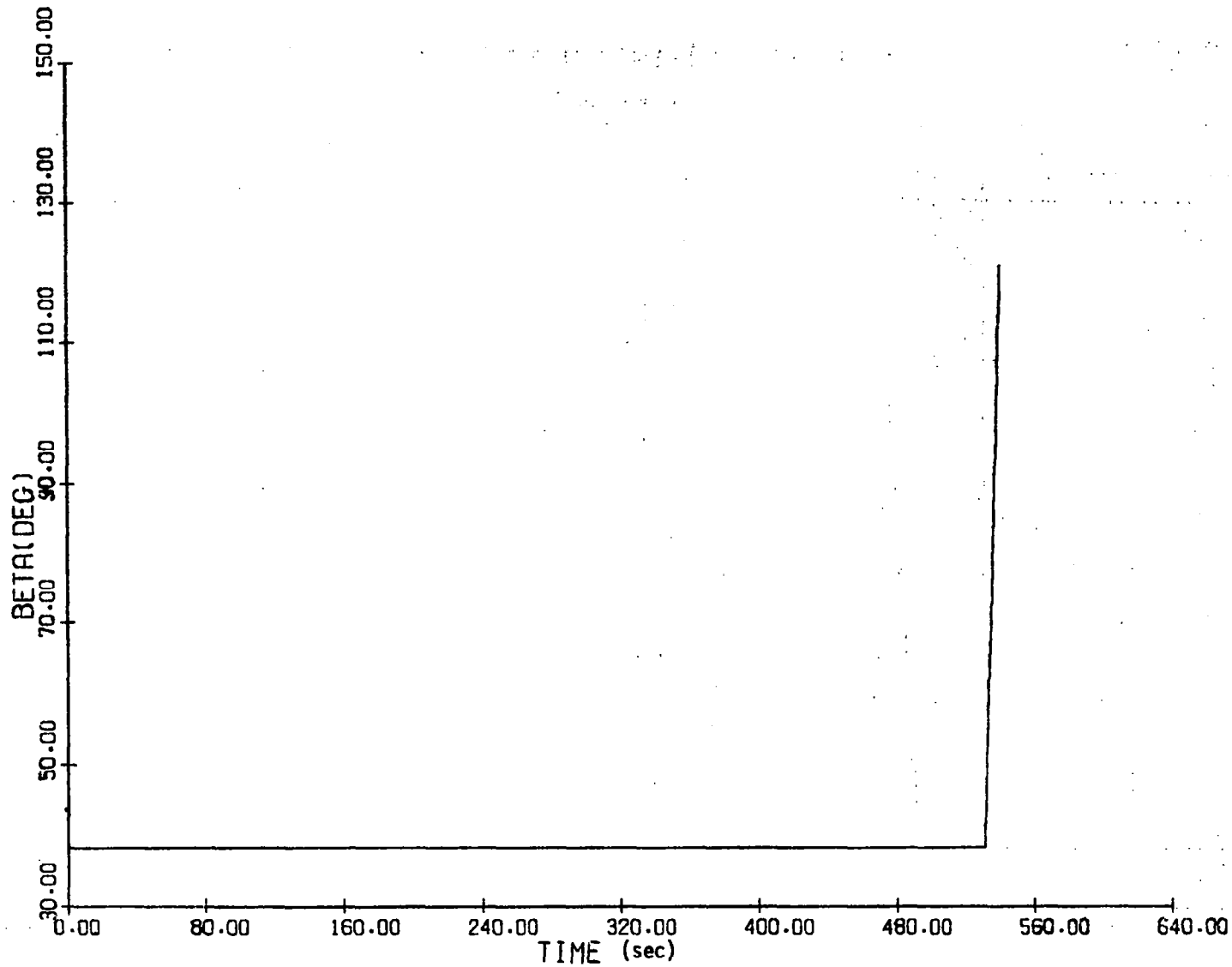


Figure 6.3.5 Changes in Heading Angle with Time

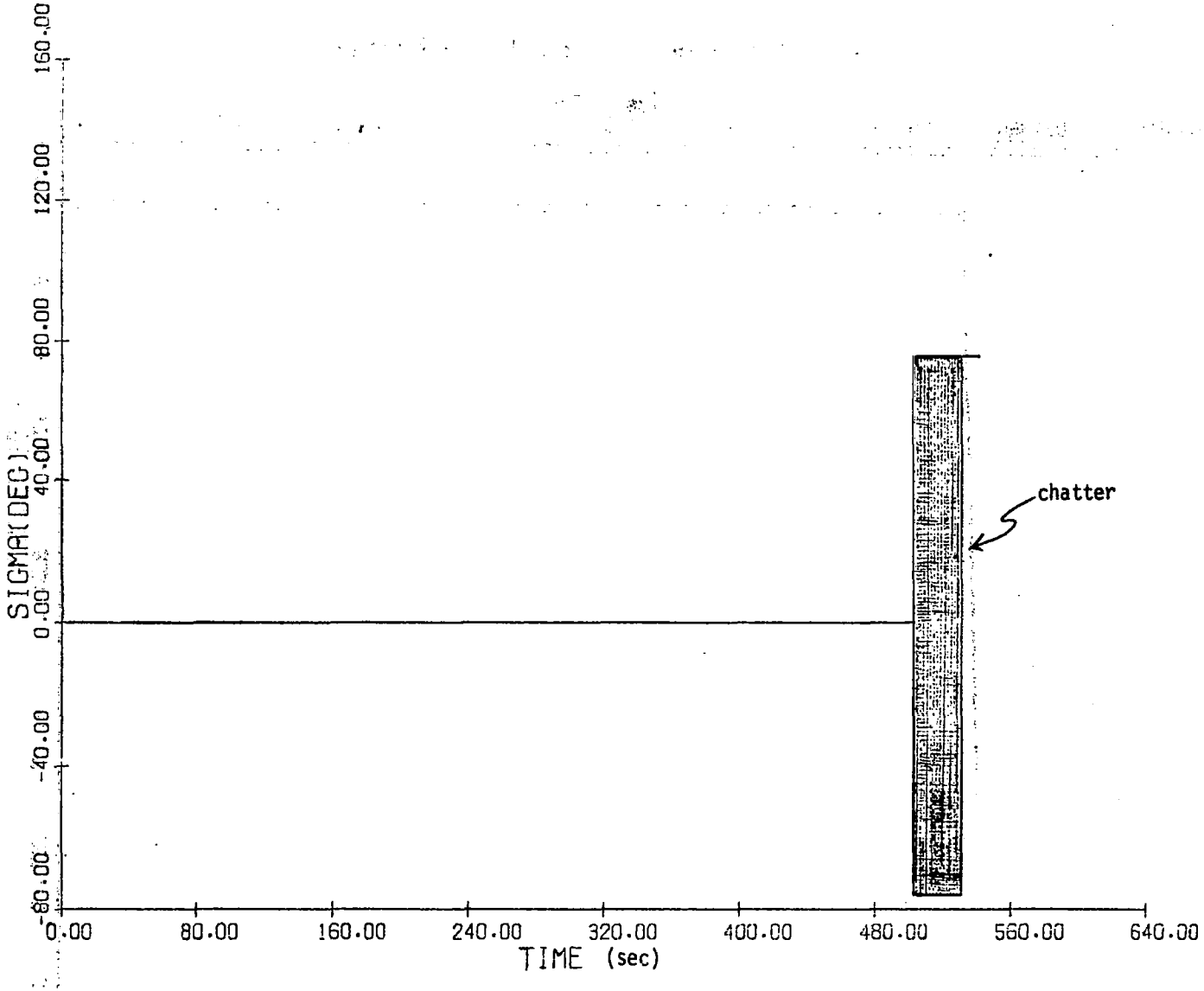


Figure 6.3.6 Bank-Angle Variations with Time

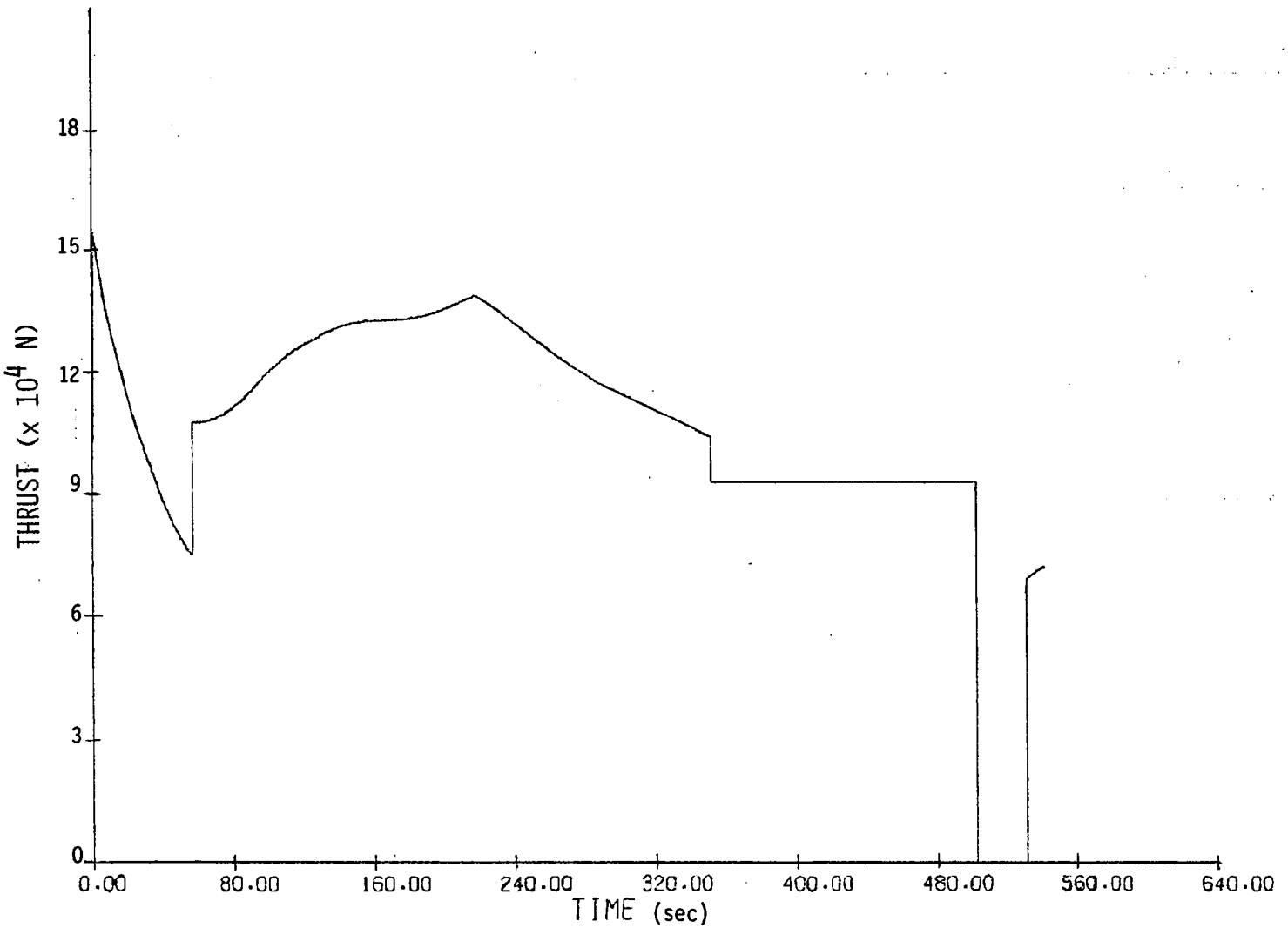


Figure 6.3.7 Changes in Thrust with Time

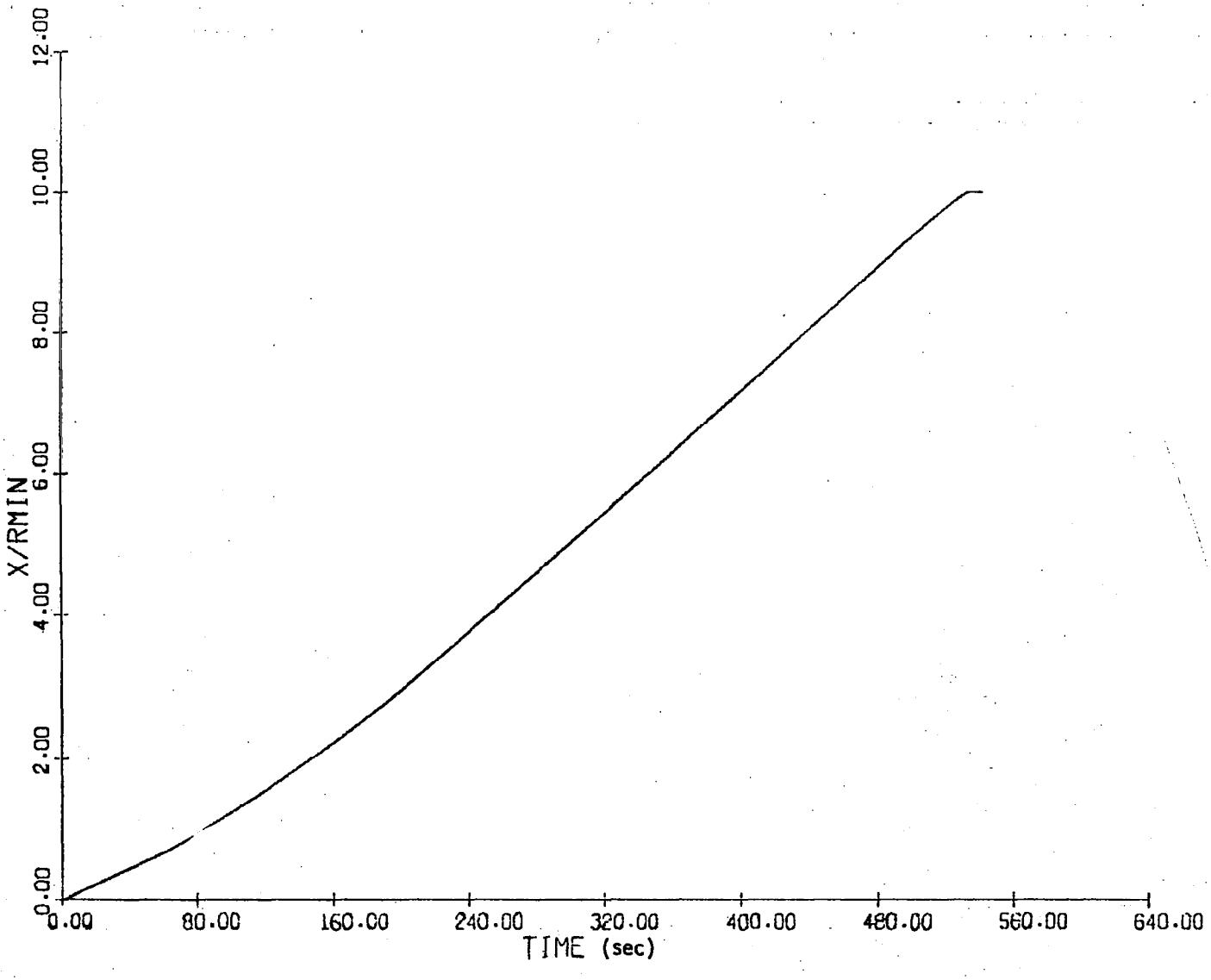


Figure 6.3.8 x-Component of Horizontal Plane Projection of Trajectory

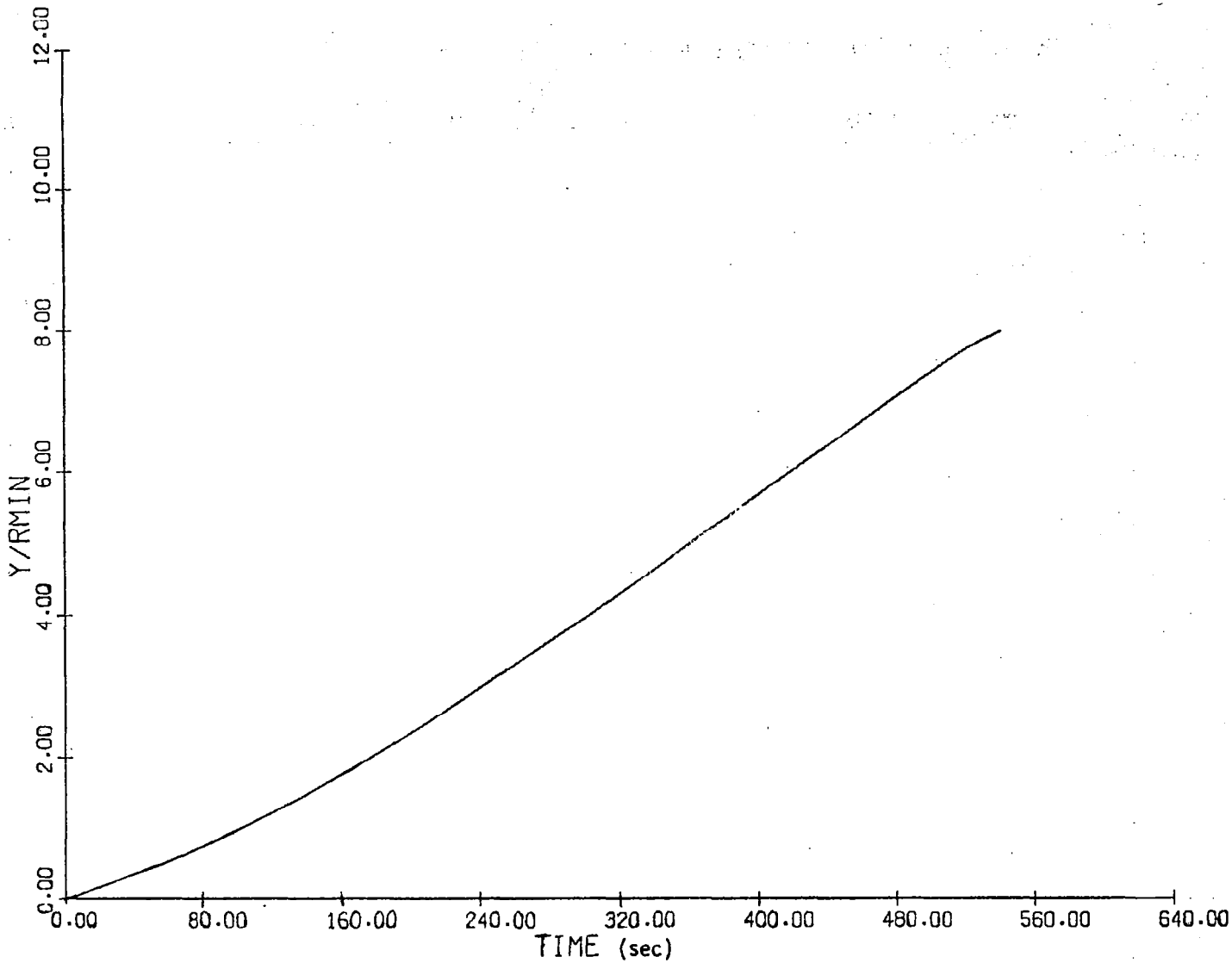


Figure 6.3.9 y-Component of Horizontal Plane Projection of Trajectory

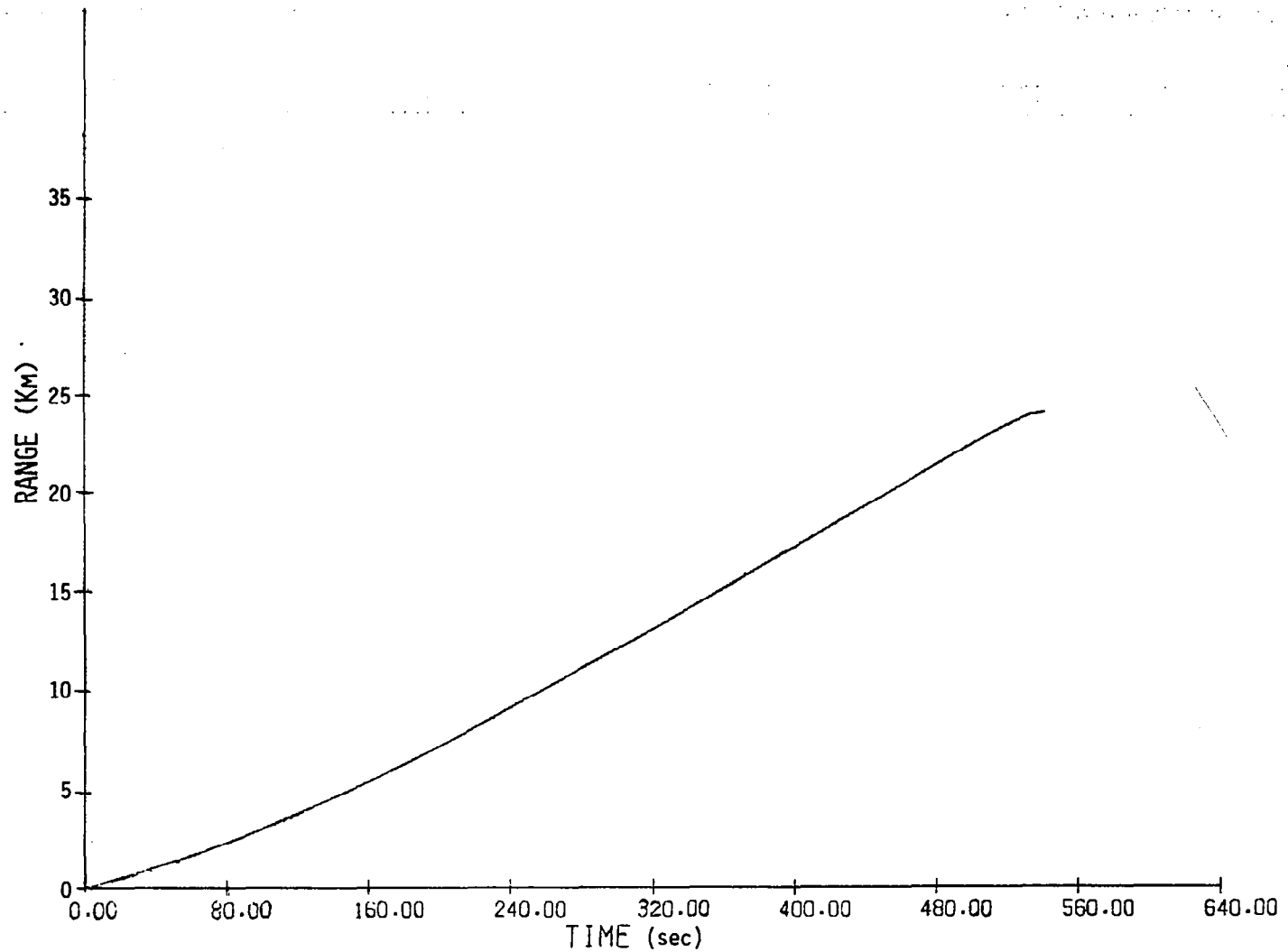


Figure 6.3.10 Range in Horizontal Plane with Time

Figure 6.3.11 Horizontal Plane Projection of Trajectory

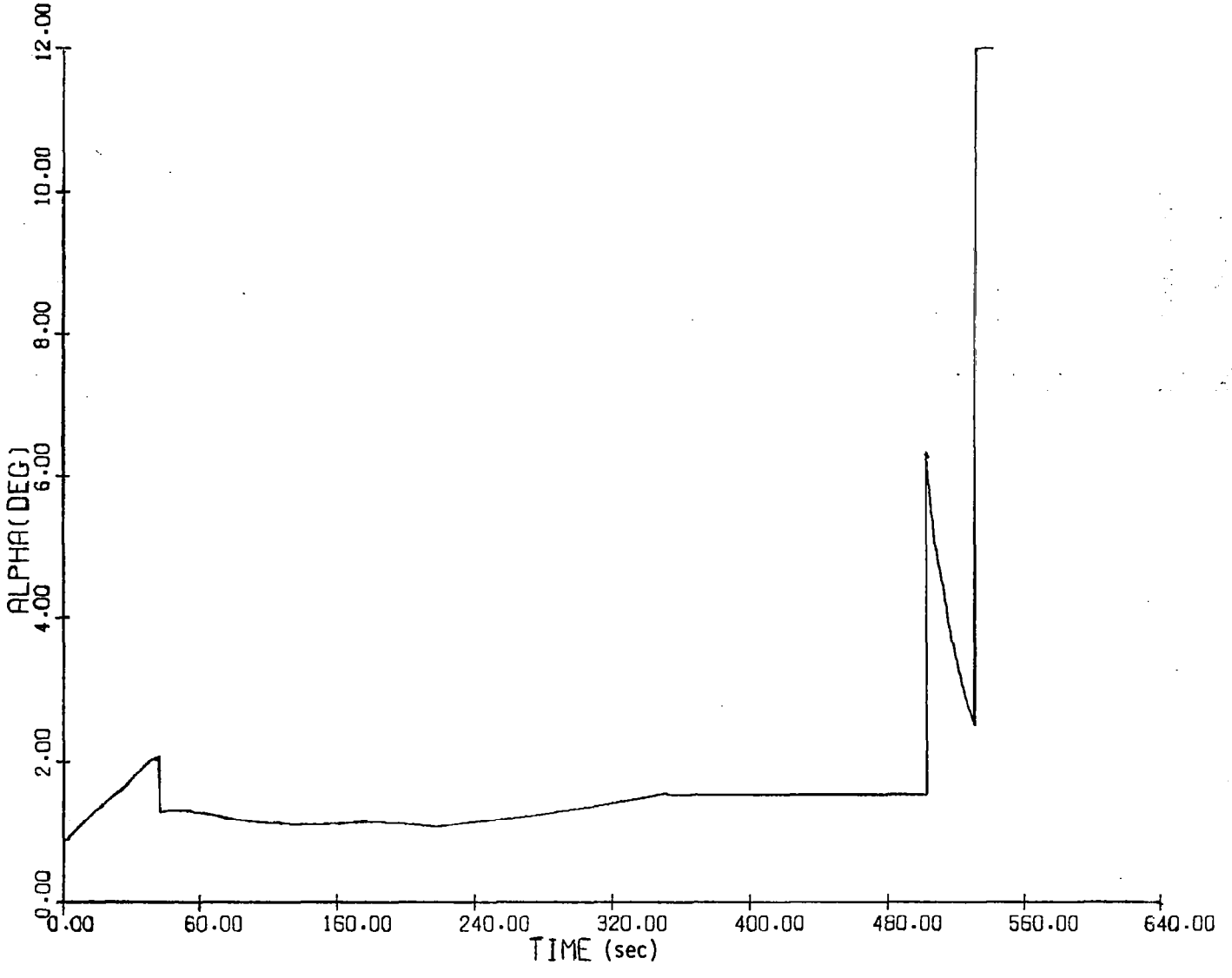


Figure 6.3.12 Alpha Time-History

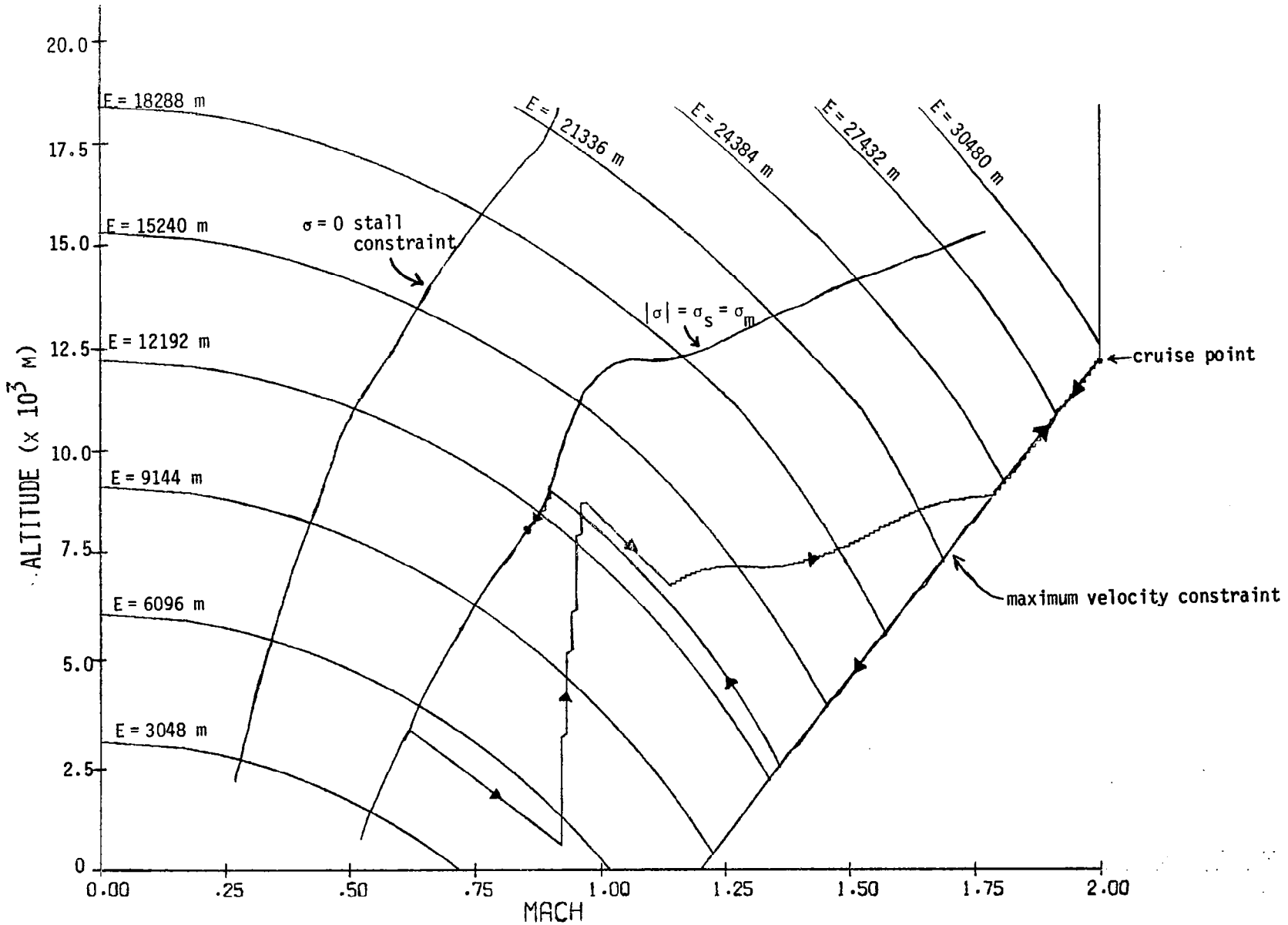


Figure 6.3.13 Altitude-Mach Number Profile of Trajectory

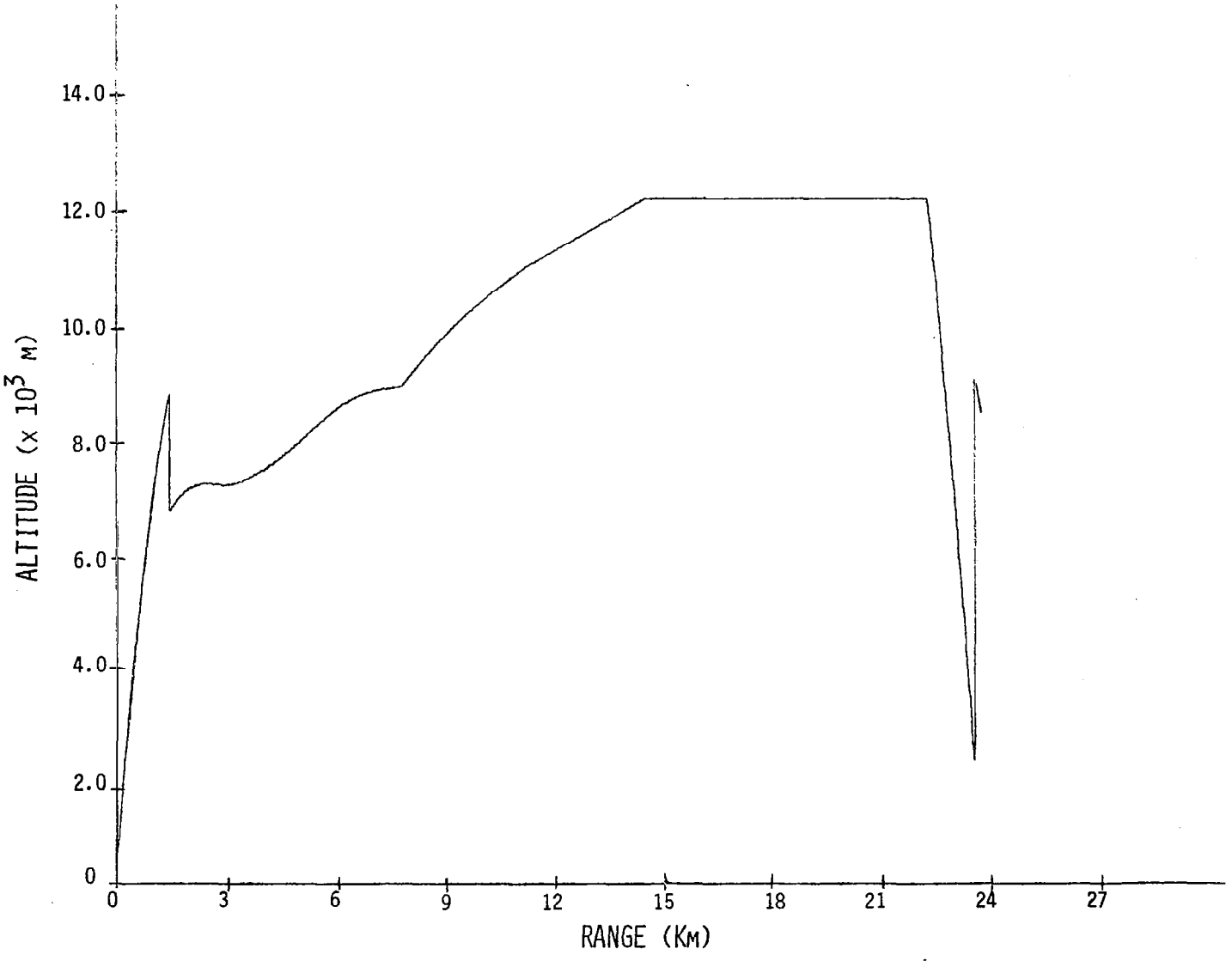


Figure 6.3.14 Altitude-Range Profile

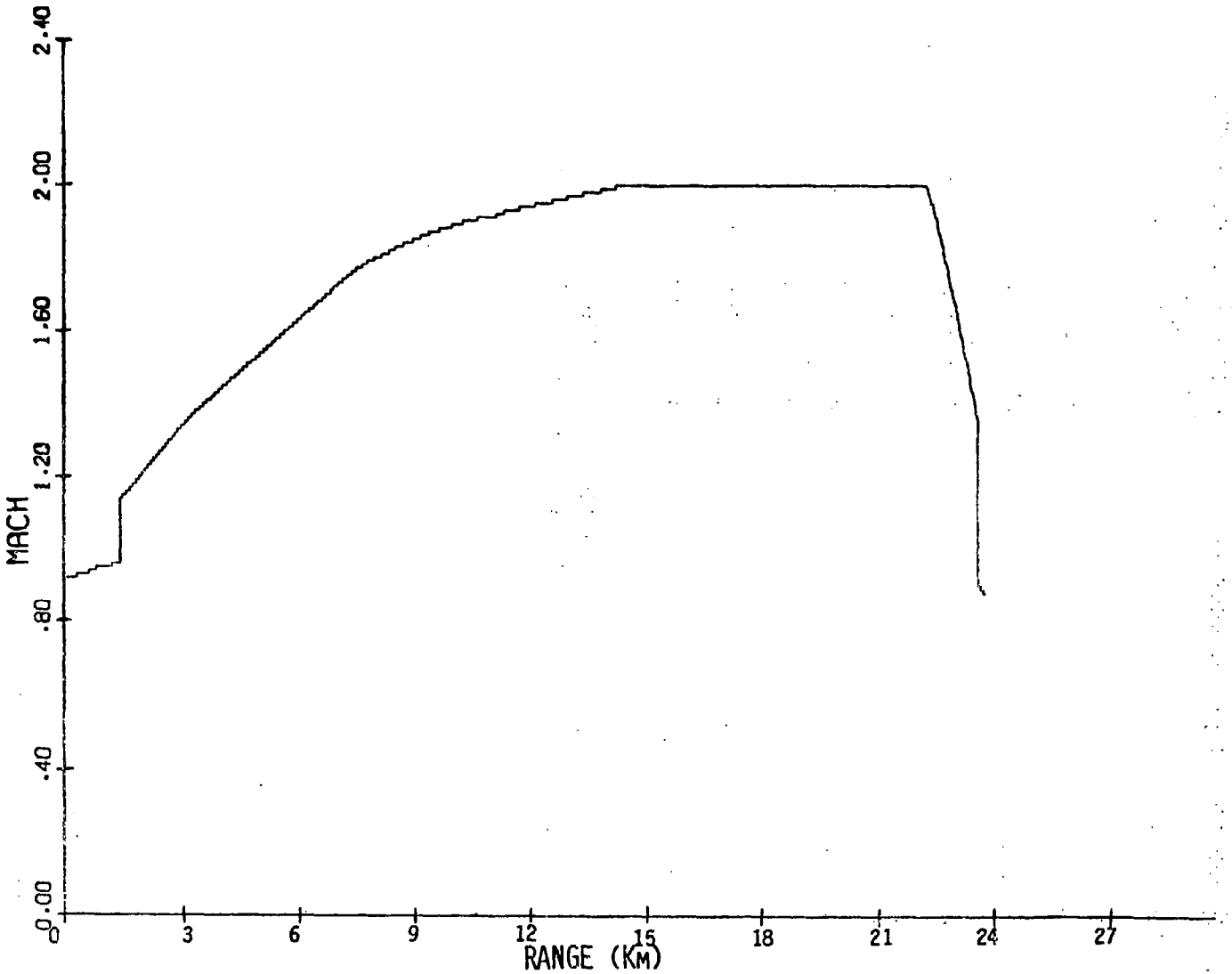
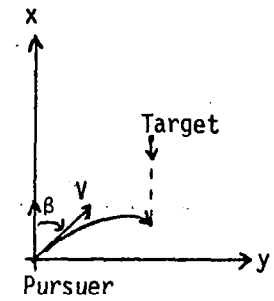


Figure 6.3.15 Variations in Mach Number with Range



Pursuer	$E_0 = 6.706\times10^3$ m	$E_f = 1.707\times10^4$ m
	$h_0 = 3.176\times10^3$ m	$h_f = 9.144\times10^3$ m
	$M_0 = 0.8$	$M_f = 1.3$
	$\gamma_0 = 0°$	$\gamma_f = 0°$
	$x_0 = 0$ Km	$x_f = 46.6$ Km
	$y_0 = 0$ Km	$y_f = 212.0$ Km
	$\beta_0 = 0°$	$\beta_f = 180°$
Target	$E_0 = 1.600\times10^4$ m	$E_f = 1.600\times10^4$ m
	$h_0 = 9.144\times10^3$ m	$h_f = 9.144\times10^3$ m
	$M_0 = 1.21$	$M_f = 1.21$
	$\gamma_0 = 0°$	$\gamma_f = 0°$
	$x_0 = 212.0$ Km	$x_f = 46.6$ Km
	$y_0 = 212.0$ Km	$y_f = 212.0$ Km
	$\beta_0 = 180°$	$\beta_f = 180°$

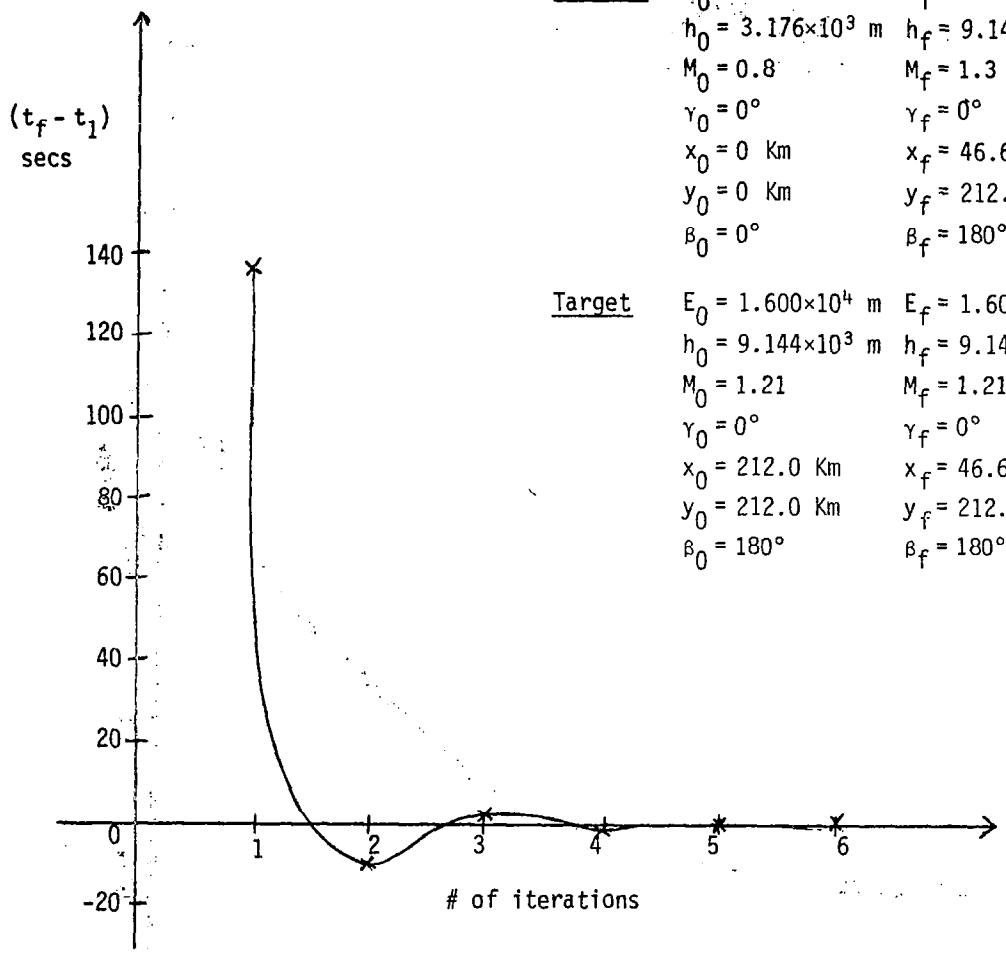
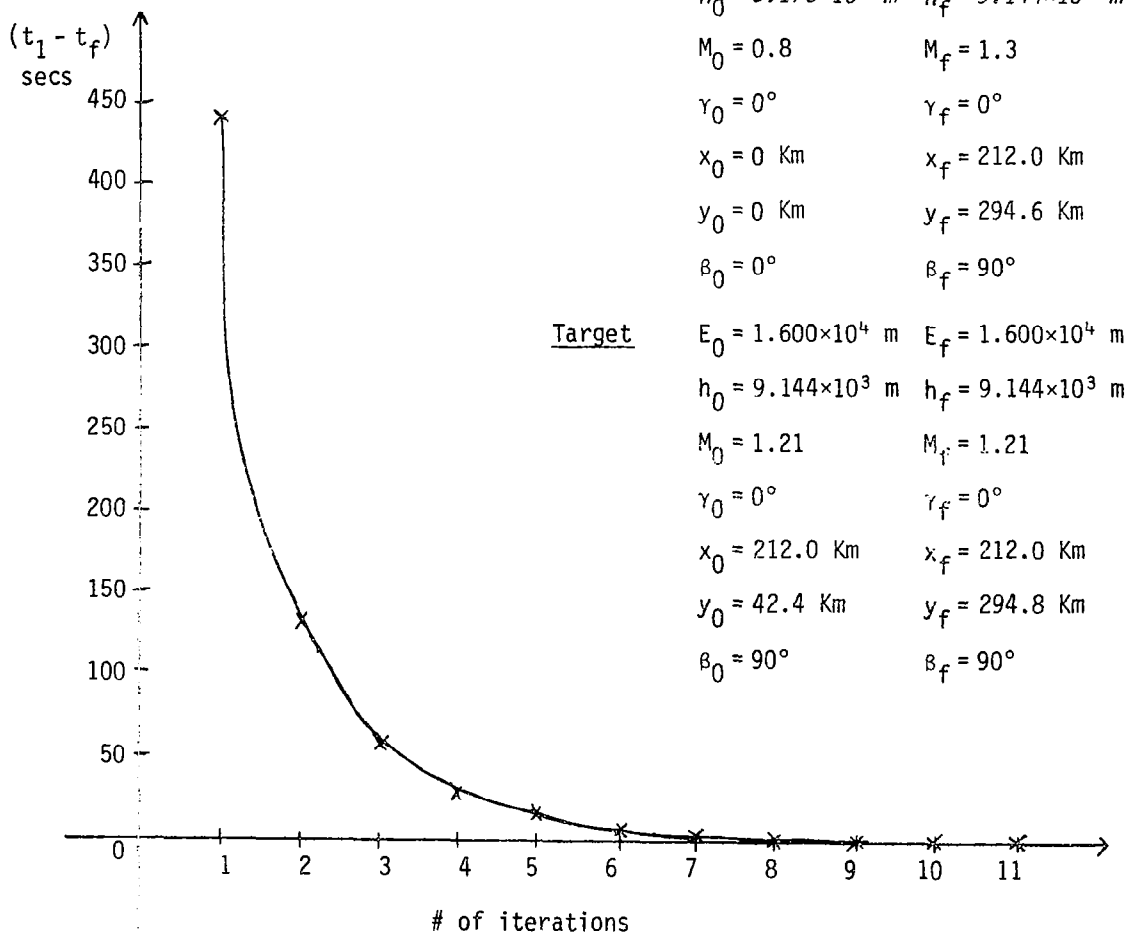


Figure 6.3.16 Convergence of t_f Iterations (180° turn)



Pursuer	$E_0 = 6.706\times10^3$ m	$E_f = 1.707\times10^4$ m
	$h_0 = 3.176\times10^3$ m	$h_f = 9.144\times10^3$ m
	$M_0 = 0.8$	$M_f = 1.3$
	$\gamma_0 = 0°$	$\gamma_f = 0°$
	$x_0 = 0$ Km	$x_f = 212.0$ Km
	$y_0 = 0$ Km	$y_f = 294.6$ Km
	$\beta_0 = 0°$	$\beta_f = 90°$
Target	$E_0 = 1.600\times10^4$ m	$E_f = 1.600\times10^4$ m
	$h_0 = 9.144\times10^3$ m	$h_f = 9.144\times10^3$ m
	$M_0 = 1.21$	$M_f = 1.21$
	$\gamma_0 = 0°$	$\gamma_f = 0°$
	$x_0 = 212.0$ Km	$x_f = 212.0$ Km
	$y_0 = 42.4$ Km	$y_f = 294.8$ Km
	$\beta_0 = 90°$	$\beta_f = 90°$

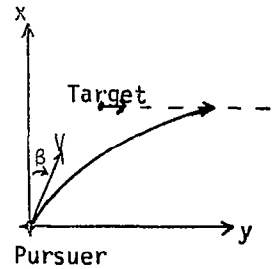
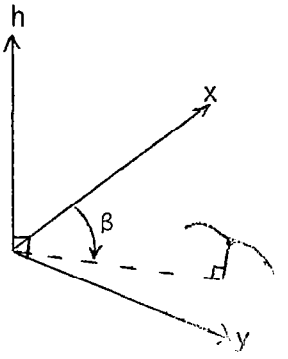


Figure 6.3.17 Convergence of t_f Iterations (90° turn)

6.4 Summary of Numerical Simulation Results

Table 6.4.1 Case 1 - Full Trajectory Simulation Using SPT Algorithms



Initial States	Final States
$E_0 = 5.483 \times 10^3$ m	$E_f = 1.067 \times 10^4$ m
$h_0 = 0.594 \times 10^3$ m	$h_f = 9.144 \times 10^3$ m
$M_0 = 0.92$	$M_f = 0.57$
$\gamma_0 = 0°$	$\gamma_f = 0°$
$x_0 = 0$ Km	$x_f = -2.03$ Km
$y_0 = 0$ Km	$y_f = 277.8$ Km
$\beta_0 = 0°$	$\beta_f = 180°$

Simulation Results

The six state simulation was done in 3 parts as below with total (minimum) time = 614.0 secs using the SPT algorithms of Section 6.2.

Part	Optimal Control Law	Minimum Time (sec)
I. Climb to cruise arc	Case 9 of Table 6.2.1. $\Delta x = 0.43$ Km, $\Delta y = 180.7$ Km Figures 6.2.15 - 6.2.30	399.3
II. Minor Corrections to achieve cruise values and cruise portion	See Section 6.2.2. $\Delta x = 0$, $\Delta y = 64.4$ Km	109.3
III. Descent from cruise arc	Case 4 of Table 6.2.3 $\Delta x = -2.46$ Km, $\Delta y = 32.7$ Km Figures 6.2.31 - 6.2.43	105.4

Table 6.4.2 Case 2 - Climb to Cruise Arc Simulation Using SPT Algorithm

Initial States	Final States
Same as in Case 1	$E_f = 2.995 \times 10^4$ m
(Table 6.4.1)	$h_f = 1.219 \times 10^4$ m
	$M_f = 2.00$
	$\gamma_f = 0°$
	$x_f = 0.43$ Km
	$y_f = 190.78$ Km
	$\beta_f = 90°$

Simulation Results

This is a six state simulation and consists of Part 1 of Table 6.4.1 plus "the minor corrections to achieve cruise values" of Part 2 of Table 6.4.1 which took 17.2 secs and covered 10.04 Km. Hence total (minimum) time = 416.5 secs. The SPT algorithm of Section 6.2 was used.

Table 6.4.3 Case 3 - Descent from Cruise Arc Simulation Using SPT Algorithm ($M_f = 0.56$)

Initial States	Final States
$E_0 = 2.995 \times 10^4$ m	$E_f = 1.067 \times 10^4$ m
$h_0 = 1.219 \times 10^4$ m	$h_f = 9.144 \times 10^3$ m
$M_0 = 2.00$	$M_f = 0.56$
$\gamma_0 = 0°$	$\gamma_f = 0°$
$x_0 = 0$ Km	$x_f = 32.7$ Km
$y_0 = 0$ Km	$y_f = 2.46$ Km
$\beta_0 = 0°$	$\beta_f = 90°$

Simulation Results

This six state simulation consists of Part 3 of Table 6.4.1 taking 105.4 secs and uses the SPT algorithm of Section 6.2.

Table 6.4.4 Case 4 - Descent from Cruise Arc Simulation Using SPT Algorithm ($M_f = 0.89$)

Initial States	Final States
Same as in	$E_f = 1.280 \times 10^4$ m
Case 3	$h_f = 9.144 \times 10^3$ m
(Table 6.4.3)	$M_f = 0.89$
	$\gamma_f = 0°$
	$x_f = 37.4$ Km
	$y_f = 2.46$ Km
	$\beta_f = 90°$

Simulation Results

This six state simulation uses the same algorithm used in Table 6.4.3 and is described in detail as Case 4 of Table 6.2.2. Total (minimum) time is 99.85 secs for a range of 37.5 Km.

Table 6.4.5 Case 5 - Target Interception (180° turn)
Using the Energy-State Approximation

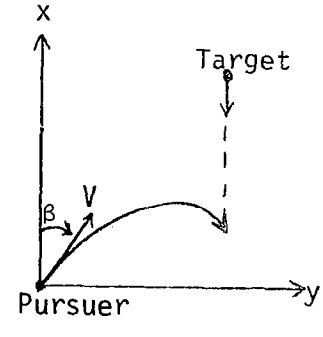
Pursuer States

$E_0 = 6.706 \times 10^3$ m	$E_f = 1.707 \times 10^4$ m
$h_0 = 3.176 \times 10^3$ m	$h_f = 9.144 \times 10^3$ m
$M_0 = 0.8$	$M_f = 1.3$
$\gamma_0 = 0°$	$\gamma_f = 0°$
$x_0 = 0$ Km	$x_f = 46.6$ Km
$y_0 = 0$ Km	$y_f = 212.0$ Km
$\beta_0 = 0°$	$\beta_f = 180°$

Target States

$E_0 = 1.600 \times 10^4$ m	$E_f = 1.600 \times 10^4$ m
$h_0 = 9.144 \times 10^3$ m	$h_f = 9.144 \times 10^3$ m
$M_0 = 1.21$	$M_f = 1.21$
$\gamma_0 = 0°$	$\gamma_f = 0°$
$x_0 = 212.0$ Km	$x_f = 46.6$ Km
$y_0 = 212.0$ Km	$y_f = 212.0$ Km
$\beta_0 = 180°$	$\beta_f = 180°$

Simulation Results



This simulation uses the energy-state approximation (4 state model of Section 5.5). The algorithm of Section 6.3 is used to iterate on t_f which is the time to interception. Six iterations were required (Figure 6.3.16) for convergence to an accuracy of 0.1 secs to yield a minimum-time trajectory taking 452.1 secs to interception. The trajectory reflects the energy-state approximation by the zoom maneuvers in altitude and velocity--the whole trajectory is similar in character to that represented by Figures 6.3.1 - 6.3.15 which used different initial and final conditions from those stated above.

Table 6.4.6 Case 6 - Target Interception (90° turn)
Using the Energy-State Approximation

Pursuer States

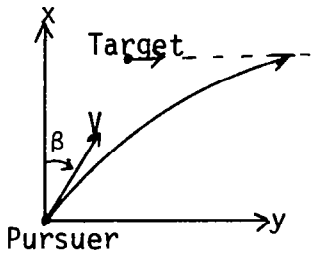
$E_0 = 6.706 \times 10^3$ m
$h_0 = 3.176 \times 10^3$ m
$M_0 = 0.8$
$\gamma_0 = 0°$
$x_0 = 0$ Km
$y_0 = 0$ Km
$\beta_0 = 0°$

$E_f = 1.707 \times 10^4$ m
$h_f = 9.144 \times 10^3$ m
$M_f = 1.3$
$\gamma_f = 0°$
$x_f = 212.0$ Km
$y_f = 294.6$ Km
$\beta_f = 90°$

Target States

$E_0 = 1.600 \times 10^4$ m
$h_0 = 9.144 \times 10^3$ m
$M_0 = 1.21$
$\gamma_0 = 0°$
$x_0 = 212.0$ Km
$y_0 = 42.4$ Km
$\beta_0 = 90°$

$E_f = 1.600 \times 10^4$ m
$h_f = 9.144 \times 10^3$ m
$M_f = 1.21$
$\gamma_f = 0°$
$x_f = 212.0$ Km
$y_f = 294.8$ Km
$\beta_f = 90°$



Simulation Results

This simulation is similar to that of Table 6.4.5 in that the energy-state approximation is used and the same algorithm (Section 6.3) is also used but differs in the initial and final states. 11 iterations were required (Figure 6.3.17) and the minimum-time trajectory takes 690.3 secs to interception.

CHAPTER 7
CONCLUSIONS

This report has described work performed on the development of a hierarchical real-time algorithm for optimal three-dimensional aircraft maneuvers using Singular Perturbation Theory (SPT). New theoretical results justify and develop systematic methods for real-time computation of nonlinear feedback controls by means of SPT and provide an assessment of the accuracy of the resulting SPT control. Practical results apply SPT to obtain a real-time feedback law for the three-dimensional minimum time long range intercept problem for an F-4 aircraft model (six state, three control variable, point mass model). Nonlinear feedback laws are presented for computing the optimal control variables u (throttle), σ (bank angle) and α (angle-of-attack) as a function of target and pursuer aircraft states and desired terminal conditions. The SPT control law results in a hierarchical nonlinear feedback structure. It is supplemented by predictive feedback terms for small deviations from the optimal trajectory and for maneuvers near the terminal time where the SPT approximation is not valid. The F-4 simulation results using the SPT control law show minor sacrifice in accuracy over the off-line optimization results, in the long range intercept case.

A real-time capability assessment of the SPT algorithm on a microcomputer has been performed and based on the results presented in this report, it may be concluded that real-time, three-dimensional long range aircraft trajectory optimization is possible using SPT. The implementation of this algorithm on a microcomputer is estimated to result in a control update cycle time of 27 msec, which is almost four times smaller than the common radar sampling interval of 100 msec. The storage and computational requirements of the algorithm are found to be well suited for on-board real-time implementation on a microcomputer.

The accuracy of the SPT solution is analyzed and it is shown how "continuation-type" methods may be used to obtain exact optimal trajectories starting from the SPT solution. The advantage of using predictive terms to supplement the SPT feedback laws is demonstrated for the aircraft trajectory optimization problem. In particular, it is shown that the SPT approximation breaks down near the terminal phase and must be corrected by "continuation" and Generalized Multiple Scale (GMS) methods.

REFERENCES

Aggarwal, R., A. J. Calise and F. Goldstein (1977), "Singular Perturbation Analysis of Optimal Flight Profiles for Transport Aircraft," Proc. Joint Auto. Cont. Conf., pp. 1261-1269.

Ardema, M. D. (1976), "Solution of the Minimum Time-to-Climb by Matched Asymptotic Expansions," J. AIAA, Vol. 14, pp. 843-850.

Ardema, M. D. (1978), "Nonlinear Singularly Perturbed Optimal Control Problems with Singular Arcs," Preprints, IFAC 7th Triennial World Congress, pp. 929-936.

Ashley, H. (1967), "Multiple Scaling in Flight Vehicle Dynamic Analysis--A Preliminary Look," AIAA paper 67-560.

Athans, M. and P. L. Falb (1966), *Optimal Control*, New York: McGraw-Hill.

Balakrishnan, A. V. (1969), "On a New Computing Technique in Optimal Control and its Application to Minimal Time Flight Profile Optimization," JOTA, Vol. 4, pp. 1-21.

Barman, J. F. and H. Erzberger (1976), "Fixed-Range Optimum Trajectories for Short-Haul Aircraft," J. Aircraft, Vol. 13, pp. 748-754.

Bellman, R. (1954), "Monotone Approximation in Dynamic Programming and the Calculus of Variations," Proc. Nat. Acad. Sci. USA, Vol. 40, pp. 1073-1075.

Bellman, R. (1957), *Dynamic Programming*, Princeton, NJ: Princeton University Press.

Bellman, R. (1961), *Adaptive Control Processes*, Princeton, NJ: Princeton University Press.

Bellman, R. and R. E. Kalaba (1965), *Quasilinearization and Nonlinear Boundary-Value Problems*, New York: Elsevier.

Binding, P. (1976), "Singularly Perturbed Optimal Control Problems I: Convergence," SIAM J. Control and Optimization, Vol. 14, pp. 591-612.

Boyd, J. R. and T. P. Christie (1965), "Energy Maneuverability Theory and Applications," 12th Annual A. F. Science and Engineering Symposium.

Breakwell, J. V. (1977), "Optimal Flight-Path-Angle Transitions in Minimum-Time Airplane Climbs," AIAA J. Aircraft, Vol. 14.

Breakwell, J. V. (1978), "More About Flight-Path-Angle Transitions in Optimal Airplane Climbs," AIAA J. Guidance and Control, Vol. 1, pp. 205-208.

Bryan, F. T. and W. A. Allison (1972), "Flight Investigation of the Optimum Energy Flight Path of an F-8D Airplane," Naval Air Test Center Technical Report No. FT-9R-72.

Bryson, A. E. and W. Denham (1962), "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. App. Mech., Vol. 29, pp. 247-257.

Bryson, A. E. (1966), "Applications of Optimal Control Theory in Aerospace Engineering," the 1966 Minta Martin Lecture, MIT.

Bryson, A. E. and M. L. Lele (1969), "Minimum Fuel Lateral Turns at Constant Altitude," AIAA J., Vol. 7, pp. 559-560.

Bryson, A. E., M. N. Desai and W. C. Hoffman (1969), "Energy-State Approximation in Performance Optimization of Supersonic Aircraft," AIAA J. Aircraft, Vol. 6, pp. 481-488.

Bryson, A. E. and M. G. Parsons (1971), "Constant Altitude, Minimum Time Turns to a Line and to a Point for a Supersonic Aircraft with a Constraint on Maximum Velocity," Stanford University, SUDAAR Report 437.

Bryson, A. E. and Y. C. Ho (1975), *Applied Optimal Control*, Washington, DC: Hemisphere Publishing Co.

Calise, A. J. (1977a), "Extended Energy Management Methods for Flight Performance Optimization," AIAA J., Vol. 15, pp. 314-321.

Calise, A. J. (1977b), "A Singular Perturbation Analysis of Optimal Thrust Control with Proportional Navigation Guidance," Proc. IEEE Conf. on Decision and Control, pp. 1167-1176.

Calise, A. J. (1978a), "Energy Management and Singular Perturbation in Flight Mechanics," Preprints IFAC 7th Triennial World Cong., pp. 949-955.

Calise, A. J. (1978b), "A New Boundary Layer Matching Procedure for Singularly Perturbed Systems," IEEE Trans. Auto. Cont., Vol. AC-23, pp. 434-438.

Campbell, T. K. and L. B. Hartsook (1972), "A New Approach to On-Board Real-Time Optimum Computations for Aerial Combat Games," J. Aircraft, Vol. 9, pp. 193-194.

Cicala, P. and A. Miele (1955a), "Branchistocronic Maneuvers of a Constant Mass Aircraft in the Vertical Plane," J. Aeronautical Sciences, Vol. 22, pp. 286-288.

Cicala, P. and A. Miele (1955b), "Branchistocronic Maneuvers of a Variable Mass Aircraft in the Vertical Plane," J. Aeronautical Sciences, Vol. 22, pp. 577-578.

Connor, M. A. (1967), "Optimization of a Lateral Turn at Constant Height," AIAA J., Vol. 5, pp. 335-338.

Davidenko, D. (1953), "On a New Method of Numerically Integrating a System of Nonlinear Equations," Dokl. Akad. Nauk. SSSR, Vol. 88, pp. 601-604 (in Russian).

Dyer, P. and S. R. McReynolds (1970), *The Computation and Theory of Optimal Control*, New York: Academic Press.

Erzberger, H. and H. Q. Lee (1971), "Optimum Horizontal Guidance Techniques for Aircraft," J. Aircraft, Vol. 8, pp. 95-101.

Ficken, F. (1951), "The Continuation Method for Functional Equations," Comm. Pure Appl. Math, Vol. 4, pp. 435-456.

Fuhrman, R. A. (1952), "Application of the Energy Concept to the Climb Performance of Turbo-Jet Propelled Airplanes," U. S. Naval Air Test Center, T.P.T. Investigative and Development Report No. 2.

Garbell, M. A. (1953), "Optimum Climbing Techniques for High-Performance Aircraft," Garbell Aeronautical Series, No. 8.

Garcia, C. G. and F. J. Gould (1977), "A Theorem on Homotopy Paths," U. Chicago Report 7734.

Glaros, L. N. and S. W. Crigler (1977), "Real-Time Trajectory Control Using Augmented Energy Management," AIAA paper 77-1052.

Hedrick, J. D. (1971), "Optimal Three-Dimensional Turning Maneuvers for Supersonic Aircraft," PhD dissertation, Stanford Univ. Dept. of Aeronautics and Astronautics.

Hedrick, J. K. and A. E. Bryson (1971), "Minimum Time Turns for a Supersonic Airplane at Constant Altitude," J. Aircraft, Vol. 8, pp. 182-187.

Hedrick, J. K. and A. E. Bryson (1972), "Three-Dimensional, Minimum-Time Turns for a Supersonic Aircraft," J. Aircraft, Vol. 9, pp. 115-121.

Heerman, H. and P. Kretsinger (1964), "The Minimum Time Problem," J. Astronautical Sciences, Vol. XI, No. 4.

Hoffman, W. C. and A. E. Bryson (1971), "A Study of Techniques for Real-Time, On-Line Optimum Flight Path Control--Minimum Time Turns to a Specified Track," Aerospace Systems, Inc., Report ASI-TR-71-4.

Hoffman, W. C. and A. E. Bryson (1973), "A Study of Techniques for Real-Time, On-Line Optimum Flight Path Control--Minimum Time Maneuvers to Specified Terminal Conditions," Aerospace Systems, Inc., Report ASI-TR-73-12.

Jamshidi, M. (1975), "Optimal Control of Nonlinear Power Systems by an Imbedding Method," Automatica, Vol. 11, pp. 633-636.

Javid, S. H. and P. V. Kokotovic (1977), "A Decomposition of Time Scales for Iterative Computation of Time-Optimal Controls," JOTA, Vol. 21, pp. 459-467.

Kaiser, A. (1944), "The Climb of Jet-Propelled Aircraft--Part 1. Speed Along The Path in Optimum Climb," Ministry of Supply RTP/T1B, Translation GDC/15/148T.

Keller, H. B. (1968), *Numerical Methods for Two-Point Boundary Value Problems*, Waltham, MA: Blaisdell.

Keller, H. B. (1977), "Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems,: in *Applications of Bifurcation Theory* (ed. P. H. Rabinowitz), New York: Academic Press.

Kelley, H. J. (1959), "An Investigation of Optimum Zoom Climb Techniques," J. Aerospace Sciences, Vol. 26, pp. 794-802.

Kelley, H. J., M. Falco and D. F. Ball (1962), "Air Vehicle Trajectory Optimization," SIAM Symp. on Multivariable System Theory, Cambridge, MA.

Kelley, H. J. (1970a), "Singular Perturbations for a Mayer Variational Problem," AIAA J., Vol. 8, p. 1177.

Kelley, H. J. (1970b), "Boundary-Layer Approximation to Powered-Flight Altitude Transients," J. Spacecraft and Rockets, Vol. 7, p. 879.

Kelley, H. J. and T. N. Edelbaum (1970), "Energy Climbs, Energy Turns and Asymptotic Expansions," J. Aircraft, Vol. 7, p. 93.

Kelley, H. J. (1971a), "Flight Path Optimization with Multiple Time Scales," J. Aircraft, Vol. 8, p. 238.

Kelley, H. J. (1971b), "Reduced-Order Modeling in Aircraft Mission Analysis," AIAA J., Vol. 9, p. 349.

Kelley, H. J. and L. Lefton (1972a), "Supersonic Aircraft Energy Turns," 5th IFAC Congress, Paris; also Automatica, Vol. 8, p. 575.

Kelley, H. J. and L. Lefton (1972b), "Differential Turns," AIAA Atmospheric Flight Mechanics Specialists Conf., Palo Alto, CA.

Kelley, H. J. (1972), "State Variable Selection and Singular Perturbations," in *Singular Perturbations: Order Reduction in Control Systems Design*, New York: ASME.

Kelley, H. J. (1973a), "3-D Energy Management for Supersonic Aircraft," 5th IFAC Symp. on Automatic Control in Space, Genoa, Italy.

Kelley, H. J. (1973b), "Some Aspects of Two-on-One Pursuit/Evasion," AIAA Aerospace Sciences Meeting; also in Automatica, Vol. 9.

Kelley, H. J. (1973c), "Aircraft Maneuver Optimization by Reduced Order Approximation," in *Control and Dynamic Systems* (ed. C. T. Leondes), New York: Academic Press.

Kellogg, R. B., T. Y. Li and J. A. Yorke (1976), "A Constructive Proof of the Brouwer Fixed Point Theorem and Computational Results," SIAM J. Numerical Analysis, Vol. 13, pp. 473-483.

Kelly, L. (1952), "Optimum Climb Technique for a Jet Propelled Aircraft," College of Aeronautics, Cranfield, Report No. 57.

Klopfenstein, R. W. (1961), "Zeroes of Nonlinear Functions," J. ACM, Vol. 8, pp. 366-373.

Kokotovic, P. V. and P. Sannuti (1968), "Singular Perturbation Method for Reducing the Model Order in Optimal Control Design," IEEE Trans. Auto. Cont., Vol. AC-13, pp. 377-384.

Kubicek, M. (1976), "Algorithm 502, 'Dependence of Solution of Nonlinear Systems on a Parameter'," ACM-TOMS, Vol. 2, pp. 98-107.

Kubicek, M. and V. Hlavacek (1972a), "Solution of Nonlinear Boundary Value Problems - Va. A Novel Method: General Parameter Mapping (GPM)," Chem. Eng. Sci., Vol. 27, pp. 743-750.

Kubicek, M. and V. Hlavacek (1972b), "Solution of Nonlinear Boundary Value Problems - Vb. Predictor-Corrector GPM Method," Chem. Eng. Sci., Vol. 27, pp. 2095-2098.

Lahaye, E. (1934), "Une Méthode de Résolution d'une Catégorie d'Équations Transcendantes," C. R. Acad. Sci. Paris, Vol. 198, pp. 1840-1842.

Lush, K. J. (1951), "A Review of the Problem of Choosing a Climb Technique with Proposals for a New Climb Technique for High Performance Aircraft," British Aeronautical Res. Council, R&M 2557.

Lush, J. H. (1956), "Optimum Climb Theory and Techniques of Determining Climb Schedules for Flight Tests," AFFTC TN-56-13.

McGill, R. and P. Kenneth (1963), "A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems," Proc. XIV, Int. Astro. Cong., Paris, Vol. 4, Paris: Gauthier-Villars.

Mehra, R. K. and R. E. Davis (1972), "A Generalized Gradient Method for Optimal Control with Inequality Constraints and Singular Arcs," IEEE Trans. Auto. Cont., Vol. AC-17, pp. 69-79.

Meier, L., C. H. Wells, S. Serebreny, D. M. Salman and J. Peschon (1970), "Information Requirements for Supersonic Transport Operations," NASA Report CR-1570.

Miele, A. (1950-51), "Problemi di Minimo Tempo nel Volo Non-Stazionario degli Aeroplani," Atti della Accademia delle Scienze di Tornio, Vol. 85, pp. 41-52.

Miele, A. (1955a), "Soluzioni Generali di Problemi di Ottimo in Volo Non-Stazionario," L'Aerotecnica, Vol. 32, pp. 135-142; also "General Solutions of Optimum Problems in Nonstationary Flight," NACA TM 1388.

Miele, A. (1955b), "Optimum Climbing Technique for a Rocket-Powered Aircraft," Jet Propulsion, Vol. 25, pp. 385-391.

Miele, A. (1958), "Flight Mechanics and Variational Problems of a Linear Type," J. Aerospace Sciences, Vol. 25, pp. 581-590.

Miele, A. (1959a), "On the Nonsteady Climb of Turbojet Aircraft," J. Aeronautical Sciences, Vol. 21, pp. 781-783.

Miele, A. (1959b), "Minimal Maneuvers of High-Performance Aircraft in a Vertical Plane," NASA Report TN D-155.

Miele, A. (1962), "Extremization of Linear Integrals by Green's Theorem," Chapter 3 of *Optimization Techniques: With Applications to Aerospace Systems* (ed. G. Leitman), New York: Academic Press.

Orava, P. J. and P. A. J. Lautala (1976), "Back and Forth Shooting Methods for Solving Two-Point Boundary Value Problems," JOTA, Vol. 18, pp. 485-498.

Orava, P. J. and P. A. J. Lautala (1977), "Interval Length Continuation Method for Solving Two-Point Boundary Value Problems," JOTA, Vol. 23, pp. 217-227.

Ortega, J. M. and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, New York: Academic Press.

Parsons, M. G. (1972), "Three Dimensional, Minimum Time Turns to a Point and Onto a Line for a Supersonic Aircraft with a Maximum Mach Number Constraint," PhD dissertation, Stanford University.

Parsons, M. G., A. E. Bryson and W. C. Hoffman (1975), "Long-Range Energy-State Maneuvers for Minimum Time to Specified Terminal Conditions," JOTA, Vol. 17, pp. 447-463.

Parsons, M. G. and A. E. Bryson (1972), "Three-Dimensional, Minimum-Time Flight Paths to a Point and Onto a Line for a Supersonic Aircraft with a Maximum Mach Number Constraint," Stanford Univ., Center for Systems Research, Report SUDAAR No. 444.

Polak, E. (1971), *Computational Methods in Optimization, A Unified Approach*, New York: Academic Press.

Ramnath, R. V. and G. Sandri (1969), "A Generalized Multiple Scales Approach to a Class of Linear Differential Equations," J. Math. Anal. Appl., Vol. 28, No. 2.

Ramnath, R. V. and P. Sinha (1975), "Dynamics of the Space Shuttle During Entry into the Earth's Atmosphere," AIAA J., Vol. 13, pp. 337-342.

Rheinboldt, W. C. (1977), "Numerical Continuation Methods for Finite Element Applications," Proc. U.S.-German Symp. on Formulation and Computational Algorithms in Finite Element Analysis (ed. Bathe), Cambridge, MA: MIT Press.

Rheinboldt, W. C. (1978), "Numerical Methods for a Class of Finite Dimensional Bifurcation Problems," SIAM J. Numerical Analysis, Vol. 15, pp. 1-11.

Roberts, S. and J. Shipman (1967), "Continuation in Shooting Methods for Two-Point Boundary Value Problems," J. Math. Anal. Appl., Vol. 18, pp. 45-58.

Roberts, S. and J. Shipman (1968), "Justification for the Continuation Method in Two-Point Boundary Value Problems," J. Math. Anal. Appl., Vol. 21, pp. 23-30.

Rudin, W. (1964), *Principles of Mathematical Analysis*, New York: McGraw-Hill.

Rutowski, E. S. (1954), "Energy Approach to the General Aircraft Performance Problem," J. Aerospace Sciences, Vol. 21, pp. 187-195.

Schneider, H. and P. B. Reddy (1974), "A New Optimization Technique for Solving Nonlinear Two-Point Boundary Value Optimal Control Problems with Application to Variable Vector Thrust Atmospheric Interceptor Guidance," AIAA paper 74-827.

Schultz, R. L. and N. R. Zagalsky (1972), "Aircraft Performance Optimization," J. Aircraft, Vol. 9, pp. 108-114.

Sederstrom, D. C., N. R. Zagalsky, R. L. Schultz, et al. (1971), "Integrated Engine Instrumentation System Study - Selected Studies on Energy Management," Honeywell, Inc., Systems and Research Division, Report No. 12591-FR2(R).

Sederstrom, D. C. (1972), "Flight Calibration Tests of F-8 Aircraft for Optimal Energy Climb," Honeywell, Inc., Systems and Research Division, Report No. 12653-FR(R).

Stein, L. H., M. L. Mathews and J. W. Frenk (1967), "STOP - A Computer Program for Supersonic Transport Trajectory Optimization," NASA Report CR-793.

Uehara, S., S. J. Homer and W. J. Lincoln (1978), "Minimum-Time Loop Maneuvers of Jet Aircraft," J. Aircraft, Vol. 15, pp. 449-455.

Vincent, T. L. and R. G. Brusch (1966), "Minimum Time Aircraft Trajectories Between Two Points in Range-Altitude Space," NASA Report CR-631.

Wasserstrom, E. (1973), "Numerical Solutions by the Continuation Method," SIAM Review, Vol. 15, pp. 89-119.

Wasow, W. (1976), *Asymptotic Expansions for Ordinary Differential Equations*, New York: Krieger.

*U.S. GOVERNMENT PRINTING OFFICE: 1979 - 635-004/15

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3167		

4. Title and Subtitle	5. Report Date
A Study of the Application of Singular Perturbation Theory	August 1979
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Raman K. Mehra, Robert B. Washburn, Salim Sajan, and James V. Carroll	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Scientific Systems, Inc. 186 Alewife Brook Parkway Cambridge, MA 02139	NAS1-15113
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Army Project No.

15. Supplementary Notes

Langley Technical Monitor: Robert S. Dunning

Final Report

16. Abstract

Work performed on the development of a hierarchical real-time algorithm for optimal three-dimensional control of aircraft is described. Systematic methods are developed for real-time computation of nonlinear feedback controls by means of singular perturbation theory. The results are applied to a six state, three control variable, point mass model of an F-4 aircraft. Nonlinear feedback laws are presented for computing the optimal control of throttle, bank angle, and angle of attack. Real time capability is assessed on a TI 9900 microcomputer. The breakdown of the singular perturbation approximation near the terminal point is examined. Continuation methods are examined to obtain exact optimal trajectories starting from the singular perturbation solutions.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
singular perturbation theory, multiple scales, continuation methods, F-4, and trajectory optimization	Unclassified - Unlimited Subject Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	338	$12.00

*For sale by the National Technical Information Service, Springfield, Virginia 22161